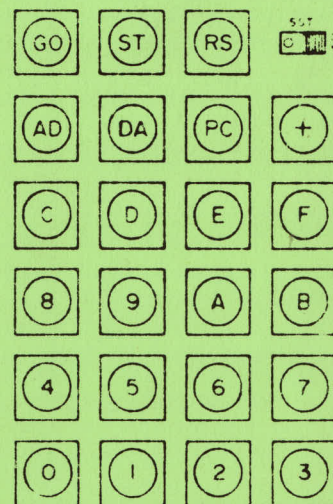


KIM GEBRUIKERS CLUB NEDERLAND  
4de Jaargang Nr. 11  
17 mei 1980



# DE KIM KENNER 11



# KIM GEBRUIKERS CLUB NEDERLAND

## Samenstelling van het bestuur:

Voorzitter	: Co Filmer Dorpsstr 1051 1566 JE ASSENDELFT Tel.: 075 - 210023
Sekretaris en ledenadministratie	: Anton Müller Sinj Semeynsstr 78 1 1061 GM AMSTERDAM Tel.: 020 - 860245
Penningmeester	: Ted Schouten Junoplnts 57 2024 RM HAARLEM Tel.: 023 - 257171 Postgirorek.nr.: 3757649
Regeling accommodatie voor KIM-club bijeenkomsten	: Bob van de Oudewetering Industriewg 12 2102 LM HEEMSTEDÉ Tel.: 023 - 286444
Technisch adviseur, cassette programma bibliotheek en propaganda KIM-club	: Uwe Schröder Echternachln 161 5625 KC EINDHOVEN Tel.: 040 - 421821
Software adviseur en regeling programma van KIM-club bijeenkomsten	: Sebo Woldringh Klieverink 619 1104 KC AMSTERDAM ZUIDOOST Tel.: 020 - 900085
Organisatie, hardware en beheer KIM-club-KIM	: Rinus Vleesch Dubois F Nightingalestr 212 2037 NG HAARLEM Tel.: 023 - 330993



# KIM INHOUDSOPGAVE

De KIM KENNER is  
een uitgave van  
de KIM Gebruikers  
club Nederland.

Adres voor het in-  
zenden van en re-  
akties op artike-  
len voor de KIM  
KENNER:

p/a H.J.C. Otten  
Dr Schaepmanstr 15  
1381 BG WEESP  
Tel.: 02940-13349

Redactie KIM KENNER:

Anton Müller  
Hans Otten  
Peter Visser

Geheel of gedeelte-  
lijke overname van  
de inhoud van de  
KIM KENNER zonder  
toestemming van  
het bestuur is ver-  
boden.

Toepassen van gepu-  
bliceerde programma's,  
hardware etc. is al-  
leen voor persoonlijk  
gebruik toegestaan.

© 1980 by KIM Gebrui-  
kers club Nederland.

Verschijnt vijf maal  
per jaar.

Pagina:

Inhoudsopgave	1
Van het bestuur	2
Van de redactie	3
Schaakprogramma, door T. Kortekaas	4
Programmeertalen - Patches op BASIC deel 2, van S.Woldringh	15
Systeemsoftware - I/O routines voor BASIC, W van Gelderen	19
Wat doe ik met mijn KIM? P.J. Visser	28
Hardware - Printer voor de KIM, Y L Bicknese	29
Hardware - Keyboard voor de KIM, Y L Bicknese	31
Hardware - RS 232 C naar TTL converter, P.J. Visser	33
Nieuws - verslag KIM-club bijeenkomst op 15 maart bij Dateq te Almere, H.J.C. Otten	34
DATA-COMMUNICATIE - een samenvatting van de door Siep de Vries gehouden lezing over datacommunicatie op 19 jan te Krommenie, P.J. Visser	35
VRAAG en AANBOD	39
AGENDA	40

# KIM VAN HET BESTUUR

MICROCOMPUTING IN 1980 MET DE 65XX PROCESSORS.

Had ik het in de vorige KIM KENNER over microcomputing in de tachtiger jaren, thans wil ik het eens hebben over microcomputing in 1980 en wat er zo al om ons heen gebeurt. Van Ir. Koopmans vernamen wij dat de continu produktie van de KIM is stopgezet en dat men deze alleen nog maar produceert in batches op het moment dat er voldoende vraag naar is. Aan de andere kant zien we nieuwe microcomputers op de markt komen die weer op de 6502 zijn gebaseerd, o.a. van ATARI (waarover d.z.z. geen nadere gegevens bekend zijn); de ESCO (Europa Single-board Computer) van Brutech Electronics uit Vinkeveen; de JUNIOR (zelfbouw) computer uit Elektuur en zo ben ik er nog een paar in de vak(hobby)-literatuur tegengekomen, waarvan ik mij de namen niet zo gauw herinner. Een source waaruit ook continu nieuwe interessante 6502 systemen rollen is Ohio Scientific Instruments (importeur ingenieursbureau Koopmans in Hardinxveld-Giessendam), zoals op de afgelopen KIM-club bijeenkomst de Challenger 4P was te zien met minifloppy en kleuren TV. Goede software voor al deze systemen is er nauwelijks en die zult U dus zelf moeten maken, of iets kopen wat in beginsel niet geschikt is voor Uw systeem, maar waarvan U gehoord of gelezen hebt dat het wel erg goed is. Een paar voorbeelden: Koopmans en/of uitgeverij De Muiderkring leveren MICRO ADE, een uitstekend produkt dat in beginsel alleen maar op een KIM met 8K extra RAM draait. (MICRO ADE is een Assembler Disassembler Editor). Een dergelijk pakket is echter in een handomdraai geïmplementeerd op elk willekeurig 6502 systeem met 8K extra RAM, er vanuit gaande dat er een character in- en output routine, alsmede een cassette save en load routine in de monitor aanwezig is. Om de kosten hoeft U het niet te laten; deze zijn + f. 100,= voor de cassette (KIM formaat) en/of + f. 100,= voor de complete source listing. Ook het implementeren van MICROSOFT BASIC (ook van Koopmans) hoeft voor de overige systemen geen probleem te zijn. Indien U daaromtrent vragen heeft belt U mij gerust, en ik help U verder. Verder krijg ik (mede door publikaties in andere bladen) de laatste tijd regelmatig vragen over PASCAL, voornamelijk van mensen die een kale KIM of AIM of SYM o.i.d. hebben. Welnu, de PASCAL compiler/interpreter die wij beschikbaar hebben voor de clubleden, is geschikt voor een KIM met minimaal 32K RAM, liever nog 40K RAM, vanwege het feit dat wij de editor van MICRO ADE gebruiken om PASCAL sources te editen en op een speciale manier op cassette te zetten. Ruwweg kost 1K RAM kant en klaar ongeveer f. 100,= zodat U al gauw een kleine vierduizend gulden aan hardware kwijt bent. De kosten van de PASCAL compiler en interpreter vallen daarbij in het niet, deze bedragen slechts f. 100,=. Overigens is deze PASCAL compiler een gestripte versie van de UCSD compiler, zonder floating point faciliteiten, hoofdzakelijk om de benodigde geheugenruimte te beperken. Voor de AIM bezitters heeft ook FAMATRA een PASCAL compiler te koop met floating point, minimale geheugen grootte 48K RAM, voor de prijs van + f. 800,= FAMATRA levert ook sinds kort een BASIC interpreter voor systeem 65 MDS, ontwikkeld door MICROSOFT, die verkrijgbaar is op een minifloppy. In Duitsland geeft ene Roland Lühr een tijdschrift uit genaamd 65XX MICRO MAG, hetgeen ik U ten zeerste kan aanbevelen. Prijs DM 50,= voor 6 nummers per jaar (eens per 2 maanden). Adres: Dipl.-Volkswirt Roland Lühr, Hansdorferstrasse 4, D-2070 Ahrensburg, Deutschland. Bestel vooral ook de backissues nr. 1 t/m 12, waaruit voor een ieder - en speciaal voor de AIM bezitters - een hoop valt te leren. Verder een bericht van Eric C. Rehnke, de uitgever van KIM (6502) USER NOTES, dat hij zijn abonneementenbestand (ongeveer 3000 abonnees) heeft verkwanzeld aan een magazine genaamd COMPUTE, dat zich eveneens voornamelijk met 65XX processoren gaat bezighouden, doch wat voornamelijk op PET/CBM en de APPLE zal zijn gericht. Iets wat ook algemeen bruikbaar is (of kan worden gemaakt) is het FIRST BOOK OF KIM, waarin een aantal interessante utilities staan die best op andere 6502 processoren kunnen worden geïmplementeerd. First Book of KIM wordt o.a. geleverd door Koopmans en 2XF in Amsterdam. Wat ook een goed boek is, is microcomputer systems principles featuring the 6502/KIM, van Camp, Smay en Triska, uitgeverij MATRIX Publishers Inc., 30NW 23rd Place, Portland, OR 97210, dat ik onlangs bij 2XF heb gekocht. Het boek vertelt ook iets over de 6800 en 8080. Voor de BASIC mensen onder ons kan ik sterk aanbevelen: Schaum's outline series, theory and problems of programming with BASIC, door Byron S. Gottfried, uitgegeven door McCraw-Hill. Een erg gemakkelijke manier om boeken te bestellen uit het buitenland, is via BOOK-IMPEX in Den Haag, kostprijs in dollars plus vijf gulden voor post en behandeling per zending. Nu we toch over boeken bezig zijn: als U interesse heeft in PASCAL, moet U op zijn minst het PASCAL User Manual and report aanschaffen (Springer-Verlag); vervolgens "An introduction to programming and problem solving with PASCAL", van Schneider, Weingart en Perlman, uitgeverij John Wiley & Sons, Inc., 605 Third Avenue, New York, NY 10016 en daarna (of misschien wel gelijktijdig): A primer on PASCAL, van Richard Conway, David Gries en E. Carl Zimmerman, op zich allemaal bolle boffen op het gebied van structured programming, uitgegeven door Winthrop Publishers, Inc., 17 Dunster Street Cambridge, Massachusetts 02138. Tot nu toe hebben wij het gehad over het gemak waarmee we diverse softwarepakketjes op de diverse 6502 systemen kunnen implementeren. Wat betreft de hardware is dat niet anders. Zo heeft bijvoorbeeld een collega van mij onlangs een IBM I/O Selectric typewriter aan zijn OSI Superboard geïnterfaced met behulp van een 6520 en nog wat andere hardware toestanden, hetgeen gemakkelijk aan iedere willekeurige 6502 processor kan worden aangesloten. In de volgende of daaropvolgende KIM KENNER kunnen we wat dat betreft wel een artikeltje verwachten. Zelf ben ik bezig met het ontwerp en de implementatie van een PIA board voor mijn KIM, bestaande uit 2 stuks 6520 met daaraan een papertape reader en een papertape punch interface, met de nodige bijbehorende software. Ook daar zal ik, als het klaar is, een artikeltje over schrijven. Dat was het dan weer voor deze keer en ik hoop dat U ook aan deze KIM KENNER weer veel plezier mag beleven.

A. Müller (sekretaris)

DE KIM KENNER



# KIM VAN DE REDACTIE

Dit is alweer het tweede nummer van de KIM KENNER in 1980 en het tweede nummer van de nieuwe redactie . De KIM KENNER levert ons veel werk op maar door de uitstekende samenwerking verloopt de productie soepel .

Ook in deze KIM KENNER is een verscheidenheid van artikelen te vinden . Drie artikelen met software , waarvan twee de I/O van de Microsoft Basic verbeteren . De I/O routines van W.v. Gelderen zijn ook voor andere software interessant. Het derde artikel met software gaat over een schaakprogramma , een fraaie prestatie om zo iets te ontwikkelen met een gewone KIM .

De hardware ontbreekt ook niet met de printerinterface en keyboard van Y.L. Bicknese .

Verder is de KIM KENNER gevuld door de redactie .

De KIM club is snel op weg een echte 6502 gebruikersclub te worden en zo treden we ook naar buiten . De publicaties in DATABUS en RADIO BULLETIN zijn daar een voorbeeld van . De club kan hierdoor groeien . Ook U als lid kan hierbij helpen . Wijst U bijvoorbeeld alle kennissen met een 6502 computer op het bestaan van de KIM club als 6502 club .

Elektuur is met een zelfbouw alternatief voor de KIM gekomen in de vorm van de Junior computer .

Duidelijk gebaseerd op de KIM ziet de Junior computer er leuk uit , en Elektuur heeft er grote plannen mee . De basis versie is nog erg beperkt maar er komen nog meer artikelen en boeken . Gebruikers en bouwers van de Junior computer zijn natuurlijk welkom in de KIM/6502 club .

H.J.C.Otten

## KIM - SCHAAKPROGRAMMA.

De doelstellingen voor dit programma waren:

Een schaak-programma te ontwikkelen, dat kon worden uitgevoerd op de standaard 1K-versie van de KIM.

Voorts dat het programma alle geldige zetten zou kunnen genereren, waaronder rochade, en-passent slaan en minor-promoties, en alle geldige zetten van de tegenpartij zou accepteren ( en ook alléén maar geldige).

Uiteraard mag van de kwaliteit van een dergelijk programma niet teveel worden verwacht, maar wellicht is het toch wel aardig voor KIM-gebruikers om met dit programma kennis te maken. Mogelijk kan dit programma als basis worden gebruikt voor verdere ontwikkelingen (maar dan wel met meer geheugen).

Als iemand belangstelling heeft om dit programma samen met mij verder te ontwikkelen, dan gaarne een berichtje of een telefoontje aan:

Theo Kortekaas,  
Kleine Poellaan 26,  
Rijzenhout.  
Tel. 02977 - 21888.

## Gebruikers - handleiding.

Er zijn twee versies van dit schaakprogramma: een versie, waarbij de computer wit speelt, en een waarbij de computer zwart speelt.

### Het laden van het programma:

Nadat de computer in gereedheid is gebracht, en de kasette-recorder is aangesloten wordt de kasette in de recorder geplaatst en voor zover nodig ge-rewind.

Op de kim-computer drukt men de volgende toetsen in:

- RS (reset)
- AD (address - selectie)
- 0, 0, F, 1 (hiermee wordt address 00 F1 geselecteerd)
- DA (data)
- 0, 0 (op address 00 F1 wordt 00 geplaatst)
- AD
- 1, 7, F, 9
- DA
- 0, 0 (op address 17 F9 wordt 00 geplaatst)
- AD
- 1, 8, 7, 3 (dit is het start-address van het laad-programma)
- GO

\* Het address is steeds op de linker vier posities van het display te zien, de data op de rechter twee posities. Na het indrukken van GO wordt het display donker.



# KIM AMUSEMENT

Hierna kan de kassette-recorder worden gestart, en wordt het programma ingelezen. Het programma bestaat uit twee delen, die afzonderlijk worden ingelezen.

Als het eerste gedeelte correct is ingelezen, verschijnt op het display: 0000 4C (inlezen eerste deel duurt  $\pm 2\frac{1}{2}$  min.)

Door nu binnen vijf seconden op de toets GO te drukken, wordt automatisch het tweede gedeelte van het programma ingelezen.

Binnen een minuut verschijnt nu op het display: 0000 EA  
Het programma is nu ingelezen en gebruiksklaar.

Als binnen een minuut het display niet oplicht, of indien op het display FFFF xx verschijnt, dan is er met het inlezen iets fout gegaan; begin opnieuw.

## Het starten van het programma en het aflezen van het display.

Nadat op het display 0000 EA is verschenen, kan het programma worden gestart. Dit geschiedt door op de toets GO te drukken.

Als de computer zwart speelt, dan verschijnt op het display: 00dE 00. De tegenspeler is nu in staat zijn eerste zet in te toetsen (Zie intoetsen zet).

Als de computer wit speelt dan begint deze na het indrukken van de toets GO aan het berekenen van de eerste zet. Het display wordt gedoofd, maar vaak licht één positie helder op.

Het berekenen van een zet duurt gemiddeld 3 minuten, maar dit kan afhankelijk van de stelling wel oplopen tot 6 minuten.

Wanneer de computer gereed is met de berekening, dan wordt het resultaat op het display vermeld.

De eerste twee posities (van links af) van het display bevatten een aanduiding van het van-veld en de derde en vierde positie een aanduiding van het naar-veld voor het stuk, dat de computer wil spelen. De normale notatie wordt gebruikt (b.v. E2 - E4).

De velden G1 t/m G8 worden aangegeven door 01 t/m 08, en de velden H1 t/m H8 door 11 t/m 18. (Dit komt, omdat de G en de H niet op een normale wijze op het display kunnen worden vertoond.)

Aanduidingen van slaan en schaak geven worden niet gegeven!

Bij En-passant slaan wordt gewoon het van en naar-veld gegeven. Uit de beweging van de pion kan wel worden afgeleid, dat het om een En-passant - situatie gaat.

Bij rochade wordt alleen de beweging van de Koning op het display getoond. Ook hier kan uit de beweging van het stuk worden opgemaakt, dat het geen gewone zet is. Tevens ziet men daaruit of het lange, dan wel korte rochade is.

Bij promotie spelen de twee rechter posities van het display een rol. Hierop verschijnt een code voor het stuk tot welk de pion is gepromoveerd.

Code	83	betekent:	Een wit paard
"	84	"	Een witte toren
"	85	"	Een witte loper
"	86	"	Een witte Dame

# KIM AMUSEMENT

"	C3	"	Een zwart paard
"	C4	"	Een zwarte toren
"	C5	"	Een zwarte loper
"	C6	"	Een zwarte Dame

Wanneer de computer geen geldige zet meer kan doen (door Pat of Mat), danwel wanneer zeer spoedig mat wordt gegeven, dan verschijnt op het display CODE xx . xx heeft hier geen betekenis. Het spel is dan afgelopen.

## Het intoetsen van een zet.

Na het displayen van een zet door de computer staat het toetsenbord geblokkeerd. Dit toetsenbord kan worden vrijgemaakt door de toets reset (RS) in te drukken.

Daarna toetst men in: AD (address - selectie) en de zet die men wil verrichten. Hiertoe kan men de gewone notatie gebruiken. (zie ook het vorige blad). Voor de G moet men de O gebruiken, voor de H de I.

Rochade en en-passant slaan worden op de zelfde manier ingegeven, als ze worden gedisplayed door de computer.

Bij promotie moet men een code meegeven om aan te geven tot welk stuk de pion promoveert. Dit kan als volgt:

- AD (address - selectie)
- O, O, O, O (address 00 00)
- DA (data invoer)
- code van het stuk (zie vorige blad)
- AD

\* in de rechter twee posities van het display is de ingegeven code te zien.

Hierna kan de zet op de normale manier worden ingegeven.

Wanneer de zet is ingegeven drukt men op de toets ST waarmee het programma weer gestart wordt voor het berekenen van de volgende zet.

Als de ingegeven zet onjuist is of ongeldig, dan verschijnt vrij snel na het indrukken van de toets ST op het display: CODE FF . Hierna kan men op de gebruikelijke manier een juiste zet ingeven.

Het mat of pat staan van de tegenspeler wordt niet afzonderlijk aangegeven. In zo'n geval is het eenvoudigweg niet meer mogelijk een nieuwe zet in te geven, en leidt elke ingegeven zet tot het verschijnen van CODE FF op het display.



## Verklaring van de gebruikte werkvelden en tabellen.

ZET1	) Interne representatie van een zet:
ZET2	) ZET1 bevat eventueel promotie-stuk.
ZET3	) ZET2 het Naar-veld en ZET3 het Van-veld.
PROM	Stuk-code als resultaat van promotie
NAAR	Naar-veld. Inhoud Hex van 00-3F
VAN	Van-veld. Inhoud Hex van 00-3F
NZET	Aantal geldige zetten per Nivo.
ITZ	Index voor tabel zetmogelijkheden.
ZWRD	Zet-waarde per nivo
STUK	Stuk-code van stuk dat gezet wordt.
ROCO	Rochade-code
WRDE	Waarde van de stelling
NIVO	Aantal halve zetten diep
CZA	Code zet aanbrengen
CZO	Code zet ongeldig
PZET	Hoogste aantal geldige zetten voor nivo 2.
OZET	Berekende aantal zetten voor nivo 2.
CZET	Code soort zet.
MAX1	Maximum aantal halve zetten diep rekenen.(normaal)
MAX2	MAXimum aantal aantal halve zetten diep rekenen voor zetten die een stuk slaan.
EPS	Veld-nummer van pion, die en-passent geslagen kan worden.
HULP	Hulp-veld
CKAZ	Code kleur aan zet.
PTZ	Pointer voor in tabel zet-mogelijkheden.
TSW	Tabel met stuk-waarden.
TZET	Tabel met zet-mogelijkheden in gecodeerde vorm.
RCT1	) Tabel met veld-nrs. die een rol spelen bij het
RCT2	) bepalen van rochade-mogelijkheden.
WIS	Wissel, die het stadium van de berekening aangeeft. (Zet aanbrengen van de tegenpartij, of berekende zet aanbrengen van eigen kleur.)
BORD	Weergave van het schaakbord. 64 posities, genummerd van hex 00 tot hex 3F. Pos. 00 = a1 01 = a2 enz. 08 = b1 09 = b2 enz. 3F = h8.
Inhoud: (hex)	
Leeg veld	00
Witte pion	80
Witte koning	82
Wit paard	83
Witte toren	84
Witte loper	85
Witte Dame	86
Zwarte pion	C1
Zwarte Koning	C2
Zwart paard	C3
Zwarte toren	C4
Zwarte loper	C5
Zwarte Dame	C6

# KIM AMUSEMENT

\*

\* WERKVELDEN EN TABELLEN.

\*

ZET1	DS	1	0000
ZET2	DS	1	0001
ZET3	DS	1	0002
PROM	DS	1	0003
NAAR	DS	1	0004
VAN	DS	1	0005
NZET	DS	1	0006
ITZ	DS	1	0007
ZURD	DS	1	0008
STUK	DS	1	0009
ROCO	DS	1	000A
LRDE	DS	1	000B
NIVO	DS	1	000C
CZA	DS	1	000D
CZO	DS	1	000E
PZET	DS	1	000F
CZET	DS	1	0010
MAX1	DS	1	0011
MAX2	DS	1	0012
EPS	DS	1	0013
HULP	DS	1	0014
CKAZ	DS	1	0015
BORD	DS	64	0016

X'FF'
X'84 83 85 86 82 85 83 84'
X'80 80 80 80 80 80 80 80'
X'00 00 00 00 00 00 00 00'
X'00 00 00 00 00 00 00 00'
X'00 00 00 00 00 00 00 00'
X'00 00 00 00 00 00 00 00'
X'00 00 00 00 00 00 00 00'
X'C1 C1 C1 C1 C1 C1 C1 C1'
X'C4 C3 C5 C6 C2 C5 C3 C4'
X'00 05 0A 13 1C 0E 0A'
X'01 01 00 03 05 03 08'
X'05 0F A4 24 00 15 1F 84'
X'34 00 06 16 22 A2 26 36'
X'A6 86 00 2E 3E AE BE 66'
X'76 E6 F6 00 06 16 22 A2'
X'00 04 07 38 3C 3F'
X'40 C0 80 10 30 20'

PTZ	DS	7	0056
TSW	DS	7	005D
TZET	DS	32	0064
RCT1	DS	6	0084
RCT2	DS	6	008A
WIS	DS	1	0090
OZET	DS	1	0091

\*

\* OVERIGE DEFINITIES

\*

DSP1	EQU	00F9
DSP2	EQU	00FA
DSP3	EQU	00FB
SCANDS	EQU	1F1F
TIME	EQU	1704

KIM - SCHAAKPROGRAMMA

AUTHOR: T. KORTEKAAS, KLEINE POELLAAN 26, RIJSENHOUT, HOLLAND.

DATE: AUGUST 1978.









# KIM AMUSEMENT

\* S R MOVE ( V E R V O L G )

```

*
*
LDA WRDE      0140 A5 08
CLC           0142 18
ADC TSW(X)    0143 75 5D
STA WRDE      0145 85 08
B245 LDA WRDE      0147 A5 08
STA ZWRD      0149 85 08
LDA MAX2      014B A5 12
CMP NIVO      014D C5 0C
BCC * + 3     014F 9D 03
JSR CALC      0151 2D 00 02
LDX X'00'     0154 A2 00
B251 PLA      0156 68
STA PROM(X)   0157 95 03
INX           0159 E8
CPX X'05'     015A E0 05
BNE B251      015C D0 F8
BIT CZ0       015E 24 0E
INC CZ0       0160 E6 0E
PLA           0162 68
BVS B268      0163 70 3E
INC NZET      0165 E6 06
BIT VAN       0167 24 05
BVS B268      0169 70 38
LDX NIVO      016B A6 0C
CPX X'01'     016D E0 01
BNE B267      016F D0 2E
BIT CZA       0171 24 0D
BVS B264      0173 70 1E
CMP ZWRD      0175 C5 08
BNE B261      0177 D0 09
JSR PTCH      0179 2D C2 00
NOP           017C EA
BNE * + 3     017D D0 03
CMP TIME      017F CD 04 17
B261 BPL B268  0182 1D 1F
LDX OZET      0184 A6 91
STX PZET      0186 86 0F
LDX X'02'     0188 A2 02
B263 LDA PROM(X) 018A 85 03
STA ZET1(X)   018C 95 00
DEX           018E CA
BPL B263      018F 1D F9
BMI B270      0191 3D 12

```

```

*
*
*
B264 LDX X'02'   0193 A2 02
B265 LDA PROM(X) 0195 85 03
CMP ZET1(X)     0197 D5 00
BNE B267        0199 D0 04
DEX             019B CA
BPL B265        019C 1D F7
BRK             019E 00

```

\* S R MOVE ( V E R V O L G )

```

*
*
B267 CMP ZWRD      019F C5 08
BMI B270          01A1 3D 02
B268 STA ZWRD      01A3 85 08
B270 LDX NAAR      01A5 A6 04
LDY BORD(X)      01A7 B4 16
PLA              01A9 68
STA BORD(X)      01AA 95 16
LDX VAN          01AC A6 05
STY BORD(X)      01AE 94 16
PLA              01B0 68
STA ROCC         01B1 85 0A
PLA              01B3 68
STA WRDE         01B4 85 08
LDY X'00'        01B6 A0 00
RTS              01B8 60

```

\* S U B R O U T I N E C A L C

```

*
*
*
CALC INC NIVO      0200 E6 0C
LDA X'CO'        0202 A9 C0
STA ZWRD         0204 85 08
EOR CKAZ         0206 45 15
STA CKAZ         0208 85 15
TYA              020A 98
STA NZET         020B 85 06
SEC              020D 38
SBC WRDE         020E E5 08
STA WRDE         0210 85 08
LDA X'3F'        0212 A9 3F
STA VAN          0214 85 05
LDX VAN          0216 A6 05
LDA BORD(X)      0218 B5 16
BEQ B088         021A F0 6C
EOR CKAZ         021C 45 15
AND X'40'        021E 29 40
BNE B088         0220 D0 66
LDA BORD(X)      0222 B5 16
AND X'07'        0224 29 07
TAX              0226 AA
LDA PTZ(X)       0227 B5 56
STA ITZ          0229 85 07
B018 LDX VAN      022B A6 05
STX NAAR         022D 86 04
B020 LSR NAAR     022F 46 04
LSR NAAR         0231 46 04
LSR NAAR         0233 46 04
TXA              0235 8A
AND X'07'        0236 29 07
TAX              0238 AA
LDY ITZ          0239 A4 07
LDA TZET(Y)      023B B9 64 00
LDY X'02'        023E A0 02

```

# KIM AMUSEMENT

\*  
\* S R C A L C ( V E R V O L G )  
\*

B021	ASL		0240	0A
	BCS	B024	0241	B0 0E
	ASL		0243	0A
	BCC	B022	0244	90 01
	INX		0246	E8
B022	ASL		0247	0A
	BCC	B023	0248	90 01
	INX		024A	E8
B023	CPX	X'08'	024B	E0 08
	BCC	B026	024D	90 0A
	BCS	B085	024F	B0 2F
B024	ASL		0251	0A
	BCC	B025	0252	90 01
	DEX		0254	CA
B025	ASL		0255	0A
	DEX		0256	CA
	BMI	B085	0257	30 27
B026	DEY		0259	88
	BEQ	B027	025A	F0 38
	STX	HULP	025C	86 14
	LDX	NAAR	025E	A6 04
	BPL	B021	0260	10 DE
B077	JSR	MOVE	0262	20 D0 00
B079	PLA		0265	68
	STA	EPS	0266	85 13
	BIT	CZ0	0268	24 0E
	BVS	B144A	026A	70 25
	LDA	ZWRD	026C	A5 08
	CMP	X'41'	026E	C9 41
	BEQ	B144A	0270	F0 1F
	LDX	VAN	0272	A6 05
	LDA	BORD(X)	0274	B5 16
	AND	X'04'	0276	29 04
	BEQ	B085	0278	F0 06
	LDX	NAAR	027A	A6 04
	LDA	BORD(X)	027C	B5 16
	BEQ	B020	027E	F0 AF
B085	INC	ITZ	0280	E6 07
	LDX	ITZ	0282	A6 07
	LDA	TZET(X)	0284	B5 64
	BNE	B018	0286	D0 A3
B088	DEC	VAN	0288	C6 05
	BPL	B010	028A	10 8A
	LDY	X'00'	028C	A0 00
	JMP	B090	028E	4C 3A 03
B144A	JMP	B144	0291	4C CE 03
B027	STA	CZET	0294	85 10
	TXA		0296	8A
	ASL		0297	0A
	ASL		0298	0A
	ASL		0299	0A

ORA	HULP	029A	05 14
STA	NAAR	029C	85 04
LDA	EPS	029E	A5 13
PHA		02A0	48
STY	PROM	02A1	84 03
BIT	CZET	02A3	24 10
BMI	* + 2	02A5	30 02
BVC	B041	02A7	50 12
STY	EPS	02A9	84 13
BPL	B040	02AB	10 4C
BVC	B077	02AD	50 83
LDA	NAAR	02AF	A5 04
BIT	CKAZ	02B1	24 15
CMP	X'20'	02B3	C9 20
BVC	B036	02B5	50 35
BCC	B079	02B7	90 AC
BCS	B037	02B9	B0 33

\*  
\* PION: SCHUIN SLAAN  
\*

B041	LDX	NAAR	02B6	A6 04
	LDA	BORD(X)	02B8	B5 16
	BEQ	B042	02BF	F0 04
	STY	EPS	02C1	84 13
	BNE	B054	02C3	D0 3A

\*  
\* PION: EN PASSENT SLAAN  
\*

B042	CPX	EPS	02C5	E4 13
	BNE	B079A	02C7	D0 68
	LDY	X'08'	02C9	A0 08
	BIT	CKAZ	02CB	24 15
B043	INX		02CD	E8
	BVS	B045	02CE	70 02
	DEX		02D0	CA
	DEX		02D1	CA
B045	DEY		02D2	88
	BNE	B043	02D3	D0 F8
	STY	EPS	02D5	84 13
	LDA	BORD(X)	02D7	B5 16
	STY	BORD(X)	02D9	94 16
	PHA		02DB	48
	TXA		02DC	8A
	PHA		02DD	48
	INC	WRDE	02DE	E6 08
	JSR	MOVE	02E0	20 D0 00
	DEC	WRDE	02E3	C6 08
	PLA		02E5	68
	TAX		02E6	AA
	PLA		02E7	68
	STA	BORD(X)	02E8	95 16
	BNE	B079A	02EA	D0 48

# THE KIM AMUSEMENT

\* PION: TWEE VELDEN VOORUIT

\*\*

B036 BCS B079A 02EC 80 46  
B037 CLC 02EE 18  
ADC VAN 02EF 65 05  
LSR 02F1 4A  
TAX 02F2 AA  
LDA BORD(X) 02F3 85 15  
BNE B079A 02F5 00 3D  
STX EPS 02F7 86 13  
B040 LDX NAAR 02F9 A6 04

PION EEN VELD VOORUIT

LDA BORD(X) 02FB 85 16  
BNE B079A 02FD 00 35

\*

\*

\*

PION: PROMOVEREN

B054 CPX X'08' 02FF E0 08  
BCC B055 0301 90 04  
CPX X'38' 0303 E0 38  
BCC B077A 0305 90 30  
B055 LDX VAN 0307 A6 05  
LDA BORD(X) 0309 85 16  
PHA 030B 48  
ORA X'03' 030C 09 03  
STA PROM 030E 85 03  
B056 LDA PROM 0310 A5 03  
LDX VAN 0312 A6 05  
STA BORD(X) 0314 95 16  
AND X'07' 0316 29 07  
CMP X'07' 0318 C9 07  
BEQ B057 031A F0 15  
TAX 031C AA  
LDA WRDE 031D A5 0B  
PHA 031F 48  
CLC 0320 18  
ADC TSW(X) 0321 75 5D  
STA WRDE 0323 85 0B  
DEC WRDE 0325 C6 0B  
JSR MOVE 0327 20 00 00  
PLA 032A 68  
STA WRDE 032B 85 0B  
INC PROM 032D E6 03  
BNE B056 032F 00 DF  
B057 PLA 0331 68  
STA BORD(X) 0332 95 16  
B079A JMP B079 0334 4C 65 02  
B077A JMP B077 0337 4C 62 02

\* TEST OP SCHAAK -  
\* STAA N

B090 LDA EPS 033A A5 13  
PHA 033C 48  
STY EPS 033D 84 13  
LDA X'40' 033F A9 40

STA VAN 0341 85 05  
STA NAAR 0343 85 04  
ASL NZET 0345 06 06  
JSR MOVE 0347 20 00 00  
LDX X'8F' 034A A2 8F  
LDA NZET 034C A5 06  
BEQ B142 034E F0 79  
LSR NZET 0350 46 06  
BCC B143 0352 90 77

\*

\*

\*

KORTE ROCHADE

JSR RC 0354 20 F0 03  
BCS B120 0357 80 30  
INX 0359 E8  
LDA BORD(X) 035A 85 16  
STX NAAR 035C 86 04  
INX 035E E8  
ORA BORD(X) 035F 15 16  
BNE B120 0361 00 26  
ASL NZET 0363 06 06  
JSR MOVE 0365 20 00 00  
LSR NZET 0368 46 06  
BCC B120 036A 90 1D  
LDX VAN 036C A6 05  
INX 036E E8  
INX 036F E8  
STX NAAR 0370 86 04  
INX 0372 E8  
LDA BORD(X) 0373 85 16  
STY BORD(X) 0375 94 16  
DEX 0377 CA  
DEX 0378 CA  
STA BORD(X) 0379 95 16  
JSR MVRO 037B 20 86 00  
LDX VAN 037E A6 05  
INX 0380 E8  
LDA BORD(X) 0381 85 16  
STY BORD(X) 0383 94 16  
INX 0385 E8  
INX 0386 E8  
STA BORD(X) 0387 95 16

\*

\*

\*

LANGE ROCHADE

B120 JSR RC 0389 20 F0 03  
ASL 038C 0A  
BCS B140 038D 80 35  
DEX 038F CA  
STX NAAR 0390 86 04  
LDA BORD(X) 0392 85 16  
DEX 0394 CA  
ORA BORD(X) 0395 15 16  
DEX 0397 CA



# KIM AMUSEMENT

ORA BORD(X)	0398	15 16	SEC	03CF	33
BNE B140	039A	00 29	SEC ZWFO	03D0	25 08
ASL NZET	039C	06 06	STA ZWRD	03D2	85 08
JSR MOVE	039E	20 00 00	TYA	03D4	98
LSR NZET	03A1	46 06	SEC	03D6	38
BCC B140	03A3	90 20	SEC WRDE	03D6	25 08
LDX VAN	03A5	A6 05	STA WRDE	03D8	85 08
DEX	03A7	CA	LDA CKAZ	03DA	A5 15
DEX	03A8	CA	EOR X'FF'	03DC	49 FF
STX NAAR	03A9	86 04	STA CKAZ	03DE	85 15
DEX	03AB	CA	DEC NIVO	03E0	06 0C
DEX	03AC	CA	LDX NIVO	03E2	A6 0C
LDA BORD(X)	03AD	B5 16	CPX X'02'	03E4	50 02
STY BORD(X)	03AF	94 16	BNE EXCL	03E6	00 04
INX	03B1	E8	LDA NZET	03E8	A5 06
INX	03B2	E8	STA NZET	03EA	85 91
INX	03B3	E8	RTS	03EC	60
STA BORD(X)	03B4	95 16			
JSR MVRD	03B6	20 B6 00			
LDX VAN	03B9	A6 05			
DEX	03BB	CA			
LDA BORD(X)	03BC	B5 16			
STY BORD(X)	03BE	94 16			
DEX	03C0	CA			
DEX	03C1	CA			
DEX	03C2	CA			
STA BORD(X)	03C3	95 16			
E I N D R O U T I N E					
C A L C					
LDX NZET	03C5	A6 06			
BNE B143	03C7	00 02			
STX ZWRD	03C9	86 08			
PLA	03CB	68			
STA EPS	03CD	A5 13			
TYA	03CE	38			

Voor computer speelt zwart te wijzigen:

Adres	0010	STY	OSP1	84	F9
	0012	JMP	CODE	4C	04 17

KIM ' SCHAAKPROGRAMMA.

Augustus 1978.

Aanvulling Febr. 1980:

Adres	00C4	CPY	X'3B'	CO	3B
-------	------	-----	-------	----	----

# KIM PROGRAMMEERTALEN

```

0010:          ***** FILE 01 *****
0020:
0030:          PATCHES BASIC DEEL 2.
0040:          -----
0050:
0060:          AUTHOR  S T WOLDRINGH
0070:                  KLIEVERINK 619
0080:                  AMSTERDAM.
0090:
0100:          DOEL VAN DE PATCHES :
0110:          1. HET CHARACTER VOOR HET DELETEN
0120:             VAN DE LAATST INGETOETSTE LETTER
0130:             IS VERANDERD VAN EEN SHIFT 0 NAAR
0140:             HET RUB-OUT (BACK-SPACE),
0150:          2. DOOR MIDDEEL VAN DE CONTROL D IS
0160:             HET NU MOGELIJK OM EEN PAGE-MODE AAN
0170:             EN AF TE ZETTEN. IN PAGE-MODE WORDT NA
0180:             IEDERE 16 REGELS GEWACHT OP EEN INPUT
0190:             ALVDRENS VERDER TE GAAN.
0200:             DE CONTROL D KAN INGETIKT WORDEN OP IEDER
0210:             WILLEKEURIG MOMENT EN HEEFT GEEN EFFECT
0220:             OP DE (VER)WERKING VAN BASIC.
0230:
0240:          BY DE PATCHES IS ER VANUIT GEGAAN
0250:          DAT DE PATCHES BASIC DEEL 1
0260:          (ZIE VORIGE KIMKENNERS) ZYN AANGEBRACHT
0270:
0280:          GEBRUIKTE VELDEN EN CONSTANTIES :
0290:
0300:          DE 00  PMODE  *      $000E
0310:          DF 00  PTEL  *      PMODE +01
0320:          EO 00  INDV  *      PTEL  +01
0330:          E2 00  INDN  *      INDV  +02
0340:          5A 1E  GETCH  *      $1E5A
0350:          A0 1E  OUTCH  *      $1FA0
0360:          69 40  RTVPL  *      $4069
0370:
0380:          04 00  CTRLD  *      $0004
0390:          0A 00  LINEFD *      $000A
0400:          10 00  AANTRG *      $0010
0410:          7F 00  DELCHR *      $007F
0420:
0010:          ***** FILE 02 *****
0020:
0030:          2437          ORG      $2437
0040:
0050:          PATCH OM TE ZORGEN DAT DE $7F
0060:          BINNEN DE GRENZEN VAN TE
0070:          AANVAARDEN CHARACTERS LIGT.
0080:
0090:          2437 C9 80          CMPIM DELCHR +01
0100:

```

# KIM PROGRAMMEERTALEN

```

0010:          ***** FILE 03 *****
0020:
0030: 243F          ORG    $243F
0040:
0050:          PATCH OM $7F ALS DELETE-CHAR
0060:          TE AANVAARDEN.
0070:
0080: 243F C9 7F      CMPIM DELCHR
0090:
0010:          ***** FILE 04 *****
0020:
0030: 2456          ORG    $2456
0040:
0050:          PATCH OM NAAR EEN NIEUWE INPUT
0060:          ROUTINE TE GAAN , DIE CHECKT
0070:          OP DE CONTROL D.
0080:
0090: 2456 20 80 17   JSR    INPUT
0100:
0010:          ***** FILE 05 *****
0020:
0030: 2A51          ORG    $2A51
0040:
0050:          PATCH OM NAAR DE NIEUWE OUTPUT-
0060:          ROUTINE TE GAAN , DIE CHECKT
0070:          OP DE REGEL TELLER EN PMODE.
0080:
0090: 2A51 20 9A 17   JSR    OUTPUT
0100:
0010:          ***** FILE 06 *****
0020:
0030: 2AE5          ORG    $2AE5
0040:
0050:          NOGMAALS NAAR DE NIEUWE INPUT-ROUTINE
0060:
0070: 2AE5 20 80 17   JSR    INPUT
0080:
0010:          ***** FILE 07 *****
0020:
0030: 4298          ORG    $4298
0040:
0050:          PATCH OM NAAR EEN OVERHEVEL-
0060:          ROUTINE TE GAAN ; DE NIEUWE INPUT-
0070:          EN OUTPUT-ROUTINE WORDEN GECODEERD
0080:          MET EEN ORG VAN $1780, ECHTER GELADEN
0090:          AAN HET EINDE VAN DE BASIC-OBJECT. BY
0100:          DE START VAN BASIC WORDT DE OBJECT
0110:          NAAR $1780 OVERGEBRACHT. (ZIE OOK
0120:          PATCHES BASIC DEEL1 WAAR HETZELFDE
0130:          GEBEURT MET DE LEES- EN SCHRIJF-ROUTINES)
0140:
0150: 4298 4C 9A 44   JMP    VRPL
0160:

```



# KIM PROGRAMMEERTALEN

```

0010:          ;          ***** FILE 08 *****
0020:          ;
0030: 449A          ORG      $449A
0040:          ;
0050:          ; ROUTINE OM DE IN- EN OUT- PUTROUTINE
0060:          ; OVER TE HEVELEN NAAR $1780; DEZE
0070:          ; OBJECT WORDT NA DE START VAN BASIC
0080:          ; VERNIETIGD (==> GEEN RUIMTE VERLIES)
0090:          ;
0100: 449A A9 CF      VRPL    LDAIM DAT      VUL HET VANAF ADRES
0110: 449C 85 E0          STAZ   INDV
0120: 449E A9 44          LDAIM DAT      /
0130: 44A0 85 E1          STAZ   INDV      +01
0140: 44A2 A9 80          LDAIM INPUT    IDEM NAAR ADRES
0150: 44A4 85 E2          STAZ   INDN
0160: 44A6 A9 17          LDAIM INPUT    /
0170: 44A8 85 E3          STAZ   INDN      +01
0180: 44AA A2 00          LDXIM $00      X=00 IVM LDAIX EN STAIX
0190: 44AC 86 DE          STXZ   PMODE    CLEAR PMODE
0200: 44AE A1 E0          VRPL1  LDAIX INDV  HAAL EEN BYTE OP
0210: 44B0 81 E2          STAIX INDN      EN ZET HET WEG BY $1780
0220: 44B2 E6 E0          INCZ   INDV      VERHOOG ADRESSEN
0230: 44B4 D0 02          BNE    VRPL2
0240: 44B6 E6 E1          INCZ   INDV      +01
0250: 44B8 E6 E2          VRPL2  INCZ   INDN
0260: 44BA D0 02          BNE    VRPL3
0270: 44BC E6 E3          INCZ   INDN      +01
0280: 44BE A9 BB          VRPL3  LDAIM ENDDT ALLES GEHAD ?
0290: 44C0 C5 E2          CMPZ   INDN
0300: 44C2 D0 EA          BNE    VRPL1      NOG NIET
0310: 44C4 A9 17          LDAIM ENDDT    /
0320: 44C6 C5 E3          CMPZ   INDN      +01
0330: 44C8 D0 E4          BNE    VRPL1      NOG STEEDS NIET
0340: 44CA A2 FF          LDXIM $FF      ALLES GEHAD, RESTORE X
0350: 44CC 4C 69 40      JMP     RTVPL     EN TERUG NAAR BASIC
0360: 44CF EA          DAT    NOP          DUMMY LABEL, HIER WORDT DE
0370:          ;          OBJECT VAN FILE 09 GEZET.
0380:          ;
0390:          ;          ***** FILE 09 *****
0400:          ;
0410: 1780          ORG      $1780
0420:          ;
0430:          ; NIEUWE INPUT EN OUTPUT ROUTINE
0440:          ; DEZE ZYN GECODEERD MET EEN ORG VAN
0450:          ; $1780 , MAAR WORDEN GELADEN ACHTER HET
0460:          ; VELD DAT.
0470:          ;
0480:          ;
0490:          ;

```

# KIM PROGRAMMEERTALEN

```

0100: 1780 20 5A 1E INPUT JSR GETCH HAAL EEN CHAR
0110: 1783 48 PHA SAVE HET OP DE STACK
0120: 1784 C9 04 CMPIM CTRLD EEN CTRL D ?
0130: 1786 D0 10 BNE INP2 NEE, DAN NIETS DOEN
0140: 1788 A5 DE LDAZ PMODE WAS ER PAGE-MODE
0150: 178A F0 06 BEQ INP1 NEE, DAN AANZETTEN
0160: 178C A9 00 LDAIM $00 JA, CLEAR HEM
0170: 178E 85 DE STAZ PMODE
0180: 1790 F0 06 BEQ INP2
0190: 1792 A9 11 INP1 LDAIM AANTRG +01
0200: 1794 85 DF STAZ PTEL RESET REGELTELLER
0210: 1796 85 DE STAZ PMODE FLAG DE PMODE
0220: 1798 68 INP2 PLA RESTORE CHAR
0230: 1799 60 RTS
0240: ;
0250: 179A C9 0A OUTPUT CMPIM LINEFD EEN LINEFEED ?
0260: 179C D0 19 BNE OUTP2 DAN GEWOON OUTPUT
0270: 179E 48 PHA SAVE DE LINEFEED
0280: 179F A5 DE LDAZ PMODE ZYN WE IN PMODE
0290: 17A1 F0 13 BEQ OUTP1 NEE DUS
0300: 17A3 C6 DF DECZ PTEL VERLAAG TELLER
0310: 17A5 D0 0F BNE OUTP1 NOG GEEN EINDE SCHERM
0320: 17A7 A9 10 LDAIM AANTRG RESET TELLER
0330: 17A9 85 DF STAZ PTEL
0340: 17AB 20 5A 1E JSR GETCH WACHT OP INPUT
0350: 17AE C9 04 CMPIM CTRLD EEN CTRL D
0360: 17B0 D0 04 BNE OUTP1 NEE DAN VERDER GAAN
0370: 17B2 A9 00 LDAIM $00 STOP DE PMODE
0380: 17B4 85 DE STAZ PMODE
0390: 17B6 68 OUTP1 PLA RESTORE DE LINEFD
0400: 17B7 20 A0 1E OUTP2 JSR OUTCH EN OUTPUT HET
0410: 17BA 60 RTS
0420: ;
0430: 17BB EA ENDDT NOP EINDE VAN ROUTINES
0440: ;

```

## SYMBOL TABLE 4000 408A

AANTRG	0010	CTRLD	0004	DAT	44CF	DELCHR	007F
ENDDT	17BB	GETCH	1E5A	INDN	00E2	INDV	00E0
INPUT	1780	INPQ	1792	INPR	1798	LINEFD	000A
OUTCH	1EA0	OUTPQ	17B6	OUTPR	17B7	OUTPUT	179A
PMODE	00DE	PTEL	00DF	RTVPL	4069	VRPL	449A
VRPLQ	44AE	VRPLR	44B8	VRPLS	44BE		

## SYMBOL TABLE 4000 408A

CTRLD	0004	LINEFD	000A	AANTRG	0010	DELCHR	007F
PMODE	00DE	PTEL	00DF	INDV	00E0	INDN	00E2
INPUT	1780	INPQ	1792	INPR	1798	OUTPUT	179A
OUTPQ	17B6	OUTPR	17B7	ENDDT	17BB	GETCH	1E5A
OUTCH	1EA0	RTVPL	4069	VRPL	449A	VRPLQ	44AE
VRPLR	44B8	VRPLS	44BE	DAT	44CF		

# KIM SYSTEM SOFTWARE

RDPF MICRO-WARE ASSEMBLER 85XX-1.0 PAGE 31

0010: 0500

0020:

0030:

0040:

0050:

0060:

0070:

0080:

0090:

0100:

0110:

0120:

0130:

0140:

0150:

0160:

0170:

0180:

0190:

0200:

0210:

0220:

0230:

0240:

0250:

0260:

0270:

0280:

0290:

0300:

0310:

0320:

0330:

0340:

0350:

0360:

0370:

0380:

0390:

0400:

0410:

0420:

0430:

0440:

0450:

0460:

0470:

0480:

0490:

0500:

0510:

0520:

0530:

0540:

RDPF ORG \$0500

\*

\* LEES EN DUMP PROGRAMMA 700H DE  
\* BASIC INTERPRETER

\*

\*

\* PROGRAMMEUR : M.V. GELDEREN

\* DATUM : 13-12-1979

\* PLAATS : KROMMENIE

\*

\* IN BASIC VERANDEREN: ADRES

\* \$2456 20 07 05

\* \$2451 20 34 25

\*

\* RESET POKE 1286,255

\* MODEM UIT POKE 1286,7

\* MODEM IN/UIT POKE 1286,8

\* PRINTER PAGINA POKE 1286,5

\* PRINTER AAN POKE 1286,4

\* TAPE LEADER POKE 1286,3

\* FILE DUMP POKE 1286,2

\* FILE READ POKE 1286,1

\* VIDEO TERMINAL POKE 1286,0

\*

0250:	F0 00	GANG	*	\$00F0	CWRITE PULSER
0260:	F1 00	TIC	*	\$00F1	CWRITE TIMER
0270:	F2 00	COUNT	*	\$00F2	CWRITE COUNTER
0280:	F3 00	IMP	*	\$00F3	TEMPORARY STORAGE
0290:	F4 00	YIMP	*	\$00F4	" "
0300:	F5 00	XTEMP	*	\$00F5	" "
0310:	FE 00	TRIR	*	\$00FE	CYCLE COUNTER
0320:	00 04	BUFFER	*	\$0400	INPUT/OUTPUT BUFFER
0330:	80 04	BUFFER1	*	\$0480	
0340:	31 20	RESTRI	*	\$2031	EDITOR WARM ENTRY ADDRESS

\* KIM ROM AND PIA ADDRESSES

0380:	42 17	SRD	*	\$1742	PIA LOCATIONN
0390:	E7 17	CHKL	*	\$17E7	CHKSUM
0400:	E8 17	CHKH	*	\$17E8	
0410:	EC 17	VER	*	\$17EC	VOLATILE EXECUTION BLOCK
0420:	F5 17	SAL	*	\$17F5	TAPE START ADDRESS
0430:	F6 17	SAH	*	\$17F6	
0440:	F7 17	EAL	*	\$17F7	TAPE END ADDRESS
0450:	F8 17	EAH	*	\$17F8	
0460:	00 14	ACIA	*	\$1400	MODEM IN/UIT
0470:	32 19	INTVER	*	\$1932	INIT VER SUBROUTINE
0480:	4C 19	CHKT	*	\$194C	CHKSUMSUBROUTINE
0490:	EA 19	INCVER	*	\$19EA	INCREMENT VER SUBROUTINE
0500:	F3 19	RDRYTE	*	\$19F3	READ BYTE SUBROUTINE
0510:	24 1A	RDCHT	*	\$1A24	READ CHAR SUBROUTINE
0520:	41 1A	RDRIT	*	\$1A41	READ PIT SUBROUTINE
0530:	8C 1E	INIT	*	\$1E8C	RESET ALL PIA'S



# KIM SYSTEM SOFTWARE

```

0010: 0500 20 04 06 STRIX JSR INOT
0020: 0503 40 05 43 JMP $4065
0030: *
0040: *
0050: 0506 20 SWITCH = $0
0060: *
0070: *
0080: 0507 AD 06 05 READ1 LDA SWITCH
0090: 050A F2 3C REQ NORM
0100: 050C 09 01 CMPIM $01
0110: 050E F0 6D REQ INALLR
0120: 0510 09 07 CMPIM $07
0130: 0512 F0 07 REQ MODI
0140: 0514 09 06 CMPIM $06
0150: 0516 FA 0C REQ MODEMI
0160: *
0170: 0518 4C 5A 1E NORM JMP $1E5A
0180: 051B 20 5A 1E MODI JSR $1F5A
0190: 051E 48 PHA
0200: 051F 20 5B 05 JSR SEE
0210: 0522 69 PLA
0220: 0523 60 RTS
0230: *
0240: 0524 AD 00 14 MODEMI LDA ACIA
0250: 0527 20 01 ANDIM $01
0260: 0529 FA F9 REQ MODEMI
0270: 052B AD 01 14 LDA ACIA +01
0280: 052E 48 PHA
0290: 052F 20 5B 05 JSR MODEMO
0300: 0532 69 PLA
0310: 0533 60 RTS
0320: *
0330: 0534 4C 06 05 WRITE1 LDY SWITCH
0340: 0537 FA 33 REQ NORMA
0350: 0539 00 02 CPYIM $02
0360: 053B FA 3D REQ OUTALA
0370: 053D 00 03 CPYIM $03
0380: 053F FA 3F REQ LIDBAR
0390: 0541 0A 04 CPYIM $04
0400: 0543 FA 6C REQ PLAT
0410: 0545 00 05 CPYIM $05
0420: 0547 FA 5A REQ PRINTR
0430: 0549 00 06 CPYIM $06
0440: 054B FA 09 REQ MODEMO
0450: 054D 00 07 CPYIM $07
0460: 054F FA 05 REQ MODEMO
0470: 0551 00 FF CPYIM $FF
0480: 0553 FA 11 REQ FIRST
0490: 0555 60 RTS
0500: *
0510: 0556 48 MODEMO PHA
0520: 0557 20 00 1E JSR $1E00
0530: 055A 68 PLA
0540: 055B 8D 01 14 SEE STA ACIA +01

```

# KIM SYSTEM SOFTWARE

```

0550: 055F A0 00 14 NMOD LDA ACIA
0560: 0561 29 02 ANDIM $02
0570: 0563 F0 F9 RFO NMOD
0580: 0565 60 RTS
0590: *
0600: 0566 A0 00 FIRST LDYIM $00
0610: 0568 80 06 05 STY SWITCH
0620: 056B 80 04 05 STY STRIX +04
0630: 056E 80 05 05 STY STRIX +05
0640: 0571 40 34 05 JMP WRITE1
0650: *
0660: 0574 20 2F 06 NORMA JSR RES
0670: 0577 40 A0 1E JMP $1E00
0680: 057A 40 68 06 OUTAIA JMP OUTALL
0690: 057D 40 DC 05 INALLR JMP INALL
0700: *
0710: 0580 A0 02 17 LIDBAR LDA $1702
0720: 0583 29 F7 ANDIM $F7
0730: 0585 80 02 17 STA $1702
0740: 0588 8E DA 05 STX MEMX
0750: 058B A2 10 LDYIM $10
0760: 058D A9 FF ONEMIA LDAIM $FF
0770: 058F 80 47 17 STA $1747
0780: 0592 A0 47 17 WACHI LDA $1747
0790: 0595 10 F5 RPL WACHI
0800: 0597 CA DEX
0810: 0598 DA F2 RNE ONEMIA
0820: 059A AE DA 05 LDY MEMX
0830: 059D 40 4D 07 JMP STPD
0840: 05A0 40 3F 06 PRTHDA JMP PRTHD
0850: *
0860: *
0870: *
0880: 05A3 C9 00 PRINTR CMPIM $00
0890: 05A5 D0 73 RNE PLXT
0900: 05A7 EF DB 05 INC PAGIN
0910: 05AA AC D3 05 PLXT LDY PAGIN
0920: 05AD C0 40 COPYIM $40
0930: 05AF F0 EF RFO PRTHDA
0940: 05B1 48 PIAT PHA
0950: 05B2 A9 7F LDAIM $7F
0960: 05B4 80 01 17 STA $1701
0970: 05B7 A9 01 LDAIM $01
0980: 05B9 00 03 17 ORA $1703
0990: 05BC 80 03 17 STA $1703
1000: 05BF F8 PIA Y0
1010: 05C0 80 00 17 STA $1700
1020: 05C3 A9 FF LDAIM $FE
1030: 05C5 20 02 17 AND $1702
1040: 05C8 80 02 17 STA $1702
1050: 05CB A9 01 LDAIM $01
1060: 05CD 00 02 17 ORA $1702
1070: 05D0 80 02 17 STA $1702
1080: 05D3 A0 00 17 DES LDA $1700

```

# KIM SYSTEM SOFTWARE

1090: 05D8 10 FB	BPL DES	
1100: 05D8 60	RTS	
1110: 05D9 00	IAVIMP =	\$00
1120: 05DA 00	MEMX =	\$00
1130: 05DB 00	PAGIN =	\$00
1140:	*	
1150:	* ROUTINE GEEFT EEN CHARACTER AAN UIT	
1160:	* EEN BUFFER EN VULT DIT BUFFER	
1170:	* ALS DEZE LEEG IS	
1180:	* CHARACTERS WORDEN VZ/E CASSETTE TAPE GELEZEN	
1190:	*	
1200:	*	
1210: 05DC 0E 0A 05	INALL STX MEMX	
1220: 05DE 05 FB	INALLA LDAAI XIEMP	
1230: 05E1 30 15	BMI RED	
1240: 05E3 04 F4	LDYZ YIMP	
1250: 05E5 00 00 00	LDAAI BUFFER	
1260: 05E8 00 00	CMPIB \$00	
1270: 05EA 00 00	BNEB PAKK	
1280: 05EC 00	PHAS	
1290: 05ED 00 FF	LDAAI \$FF	
1300: 05EF 00 FF	STAZ XIEMP	
1310: 05F1 00	PLA	
1320: 05F2 00 00	PAKK INCZ YIMP	
1330: 05F4 00 00 05	LDX \$05	
1340: 05F7 00	RTS	
1350: 05F8 00 00 06	RED JSR GETLIN	
1360: 05FA 00 00	LDAAI \$00	
1370: 05FB 00 00	STAZ YIMP	
1380: 05FD 00 00	STAZ XIEMP	
1390: 05FF 00 00	JMP INALLA	
1400:		
1410:	*	
1420:	* INITIALISERING VZ/H LEES/DUMP BUFFER	
1430:	*	
1440: 0604 00 00	INCI LDAAI \$00	
1450: 0606 00 00 17	STA \$1703	
1460: 0609 00 00	LDAAI \$00	
1470: 060B 00 00 17	STA \$1702	
1480: 060E 00 00	LDAAI \$00	
1490: 0610 00 00 14	STA ACIA	
1500: 0613 00 11	LDAAI \$11	
1510: 0615 00 00 14	STA ACIA	
1520: 0618 00 FF	LDXIM \$FF	
1530: 061A 00	PLA	
1540: 061B 00	TAY	
1550: 061C 00	PLA	
1560: 061D 00	TXS	
1570: 061E 00	PHAS	
1580: 061F 00	TYAS	
1590: 0620 00	PHA	
1600: 0621 00 00	STXZ XIEMP	
1610: 0623 00 00	LDAAI \$00	
1620: 0625 00 00 05	STA SWITCH	





```

1630: 0628 A9 00 LDAIM $00
1640: 062A E5 F4 STAZ YTMP
1650: 062C 8D D9 05 STA TAMPMP
1660: 062E A0 30 RES IDYIM $30
1670: 0631 8C EF 06 STY READ
1680: 0634 A0 31 LDYIM $31
1690: 0636 8C F0 06 STY READNO
1700: 0639 A0 3D LDYIM $3D
1710: 063B 8C D8 05 STY PAGIN
1720: 063E 60 RTS
1730:
1740:
1750:
1760: 063F A0 00 PRINT DE PAGINA KOP
1770: 0641 48 PHA
1780: 0642 39 F4 06 PLEI LDAAY IKSTX
1790: 0645 20 B1 05 JSR PLAT
1800: 0648 C8 INY
1810: 0649 C0 20 COPYIM $20
1820: 064B D0 F5 RNE PLEI
1830: 064D A9 00 LDAIM $00
1840: 064F 8D D8 05 STA PAGIN
1850: 0652 EF F0 06 INC READNO
1860: 0655 AD F0 06 LDA READNO
1870: 0658 C9 3A CMPIW $3A
1880: 065A D0 08 RNE MIN
1890: 065C A9 31 LDAIM $31
1900: 065E 8D F0 06 STA READNO
1910: 0661 EF EF 06 INC READ
1920: 0664 68 MIN PLA
1930: 0665 4C B1 05 JVP PLAT
1940:
1950:
1960:
1970:
1980:
1990:
2000: 0668 AC D9 05 OUTAIL LDY TAMPMP
2010: 066B C9 0A CMPIW $0A
2020: 066D D0 18 RNE NODUP
2030: 066F 99 80 04 STAAY BUFFER1
2040: 0672 C8 INY
2050: 0673 A9 0A LDAIM $0A
2060: 0675 99 80 04 STAAY BUFFER1
2070: 0678 8F DA 05 STY MEMX
2080: 067B 20 04 07 JSR DUMP
2090: 067E AE DA 05 LDY MEMX
2100: 0681 A9 00 LDAIM $00
2110: 0683 8D D9 05 STA TAMPMP
2120: 0686 60 RTS
2130: 0687 99 80 04 NODUP STAAY BUFFER1
2140: 068A EF D9 05 INC TAMPMP
2150: 068D 60 RTS
2160:

```

# KIM SYSTEM SOFTWARE

2170:		* IFES EFN REGEL	2590:	*
2180:		* V/D TAPE	2600:	*
2190: 068E AD 02 17	GETLIN LDA \$1702	2610:	* PAGINA KOP	
2200: 0691 29 FB	ANDIM \$FB	2620: 06E4 0D	TKSTX =	\$0D
2210: 0693 8D 02 17	STA \$1702	2630: 06E5 0A	=	\$0A
2220: 0696 A9 7F	LDAIM \$7F	2640: 06E6 2D	=	1-
2230: 0698 8D 41 17	STA \$1741	2650: 06E7 0A	=	\$0A
2240: 069A A9 13	LDAIM \$13	2660: 06E8 50	=	1P
2250: 069D 8D 42 17	STA SRD	2670: 06E9 41	=	1A
2260: 06A0 D8	CID	2680: 06EA 47	=	1G
2270: 06A1 20 AF 06	NEWLY JSR GETLYN	2690: 06EB 49	=	1I
2280: 06A4 AD 02 17	ENDAD LDA \$1702	2700: 06EC 4E	=	1N
2290: 06A7 09 04	ORAIM \$04	2710: 06ED 41	=	1A
2300: 06A9 8D 02 17	STA \$1702	2720: 06EE 20	=	1
2310: 06AC 4C 8C 1E	JMP INIT	2730: 06EF 30	RIAD =	\$30
2320:	*	2740: 06F0 01	RIADNO =	\$01
2330: 06AF A9 00	GETLYN LDAIM \$00	2750: 06F1 20	=	1
2340: 06B1 85 F4	STAZ YIMP	2760: 06F2 4D	=	1W
2350: 06B3 20 41 1A	SYNC JSR RDRIT	2770: 06F3 49	=	1I
2360: 06B6 A6 F3	LSPZ IMP	2780: 06F4 43	=	1C
2370: 06B8 25 F3	ORAZ IMP	2790: 06F5 52	=	1W
2380: 06BA 85 F3	STAZ IMP	2800: 06F6 4F	=	10
2390: 06BC 8D 40 17	STA \$1740	2810: 06F7 53	=	1S
2400: 06BF C9 16	TST CMPIM \$16	2820: 06F8 4F	=	10
2410: 06C1 D0 F3	RNE SYNC	2830: 06F9 46	=	1F
2420:	*	2840: 06FA 54	=	1T
2430: 06C3 20 24 1A	JSR RDRIT	2850: 06FB 20	=	1
2440: 06C6 8D 40 17	STA \$1740	2860: 06FC 42	=	1R
2450: 06C9 C9 13	CMPIM \$13	2870: 06FD 41	=	1A
2460: 06CB D0 F2	RNE TST	2880: 06FE 53	=	1S
2470:	*	2890: 06FF 49	=	1I
2480: 06CD A0 00	GETCHR LDYIM \$00	2900: 0700 43	=	1C
2490: 06CF 84 F4	STYZ YIMP	2910: 0701 0A	=	\$0A
2500: 06D1 20 24 1A	GET JSR RDRIT	2920: 0702 0D	=	\$0D
2510: 06D4 A4 F4	LDYZ YIMP	2930: 0703 0A	=	\$0A
2520: 06D6 C0 81	CPYIM \$81	2940:		
2530: 06D8 F0 05	REQ DOIR	2950:		
2540: 06DA 99 00 04	STAAY BUFFER			
2550: 06DD E6 F4	INC YIMP			
2560: 06DE C9 0D	DOIR CMPIM \$0D			
2570: 06E1 D0 EE	RNE GET			
2580: 06E3 60	RTS			

0210: 0704 AD 02 17 DUMP LDA \$1702 TURN ON CASSETTE

0220: 0707 29 F7 ANDIM \$F7

0230: 0709 8D 02 17 STA \$1702

0240:

0250: \* OUTPUT SOURCE DATA

0260: 070C A9 27 SORCOT LDAIM \$27

0270: 070E 85 0 STAZ \$00F0

0280: 0710 A9 AD LDAIM \$AD

0290: 0712 8D EC 17 STA VER SETUP VER

0300: 0715 A9 00 LDAIM \$00

0310: 0717 8D E7 17 STA CHKL

0320: 071A 8D E8 17 STA CHKH

0330: 071D A9 80 LDAIM BUFFER1

# KIM SYSTEM SOFTWARE

0140: 071F 8D ED 17	STA	VER	+01
0150: 0722 A9 04	LDAIM	BUFFER1	/
0160: 0724 8D EE 17	STA	VER	+02
0170: 0727 A9 60	LDAIM	\$60	
0180: 0729 8D EF 17	STA	VER	+03
0190: 072C A9 BF	LDAIM	\$BF	TURN ON OUTPUT
0200: 072E 8D 43 17	STA	\$1743	TO CASSETTE
0210: 0731 A2 64	LDXIM	\$64	SEND 100 SYNC PULSES
0220: 0733 A9 16	LEADER LDAIM	\$16	
0230: 0735 20 5F 07	JSR	HIC	
0240: 0738 A9 13	LDAIM	\$13	SEND START OF DATA CHAR
0250: 073A 20 6R 07	JSR	OUTCHT	
0260: 073D 20 EC 17	JSR	VER	NEXT
0270: 0740 20 EA 19	JSR	INCOVER	
0280: 0743 C9 0A	CMPIW	\$0A	
0290: 0745 D0 08	BNE	NOCRA	BRANCH IF NO END OF LINE
0300: 0747 20 6R 07	JSR	OUTCHT	END OF LINE
0310: 074A A2 02	LDXIM	\$02	WAIT A MOMENT
0320: 074C 4C A5 07	ONEMIN JMP	TAIL	
0330: 074F C9 40	NOCRA CMPIW	\$40	END OF FILE?
0340: 0751 F0 06	REQ	EOFC	BRANCH IF YES
0350: 0753 20 6R 07	JSR	OUTCHT	
0360: 0756 4C 3D 07	JMP	NEXT	
0370: 0759 20 6R 07	EOFC JSR	OUTCHT	
0380: 075C 4C A5 07	JMP	TAIL	
0390:	*		
0400:	* DUMP	LEADER	
0410: 075F 86 F1	HIC	STXZ	TIC
0420: 0761 48	HICA	PHA	
0430: 0762 20 6R 07	JSR	OUTCHT	
0440: 0765 F8	PLA		
0450: 0766 CF F1	DECZ	TIC	
0460: 0768 D0 F7	BNE	HICA	
0470: 076A 60	RTS		
0480:	*		
0490:	* SUR TO SEND ONE 8	RIT BYTE	
0500: 076B A0 08	OUTCHT LDYIM	\$08	
0510: 076D 84 F2	STYZ	COUNT	8 BIT COUNT
0520: 076F A0 02	TRY LDYIM	\$02	START AT
0530: 0771 84 FE	STYZ	TRIR	3600 HERTZ
0540: 0773 BE A1 07	ZON LDXAY	NPUL	NUMBER OF HALF CYCLES
0550: 0776 48	PHA		SAVE THE CHAR
0560: 0777 2C 47 17	ZONA RIT	\$1747	WAIT FOR END OF CYCLE
0570: 077A 10 F8	BPL	ZONA	
0580:			
0590: 077C B9 A2 07	LDAAY	TIMG	SET UP TIMER
0600: 077F 8D 44 17	STA	\$1744	FOR THIS PULSE
0610:			
0620: 0782 A5 F0	LDAZ	GANG	CHANGE STATE
0630: 0784 49 80	EORIM	\$80	OF OUTPUT
0640: 0786 8D 42 17	STA	\$1742	PORT
0650:			
0660: 0789 85 F0	STAZ	GANG	AND SAVE STATE
0670: 078B CA	DEX		DONE ALL CYCLES?
0680: 078C D0 E9	BNE	ZONA	NO THEN SEND ANOTHER

# KIM SYSTEM SOFTWARE

0690:					
0700:	078E	68	PLA		
0710:	078F	C6 FE	DECZ	TRIB	ONE MORE GONE
0720:	0791	F0 05	BEQ	SETZ	THE LAST ONE TOO
0730:	0793	30 07	BMI	ROUT	EVEN THE LAST ONE WENT
0740:					
0750:	0795	4A	LSRA		ANOTHER BIT TO THE CARRY
0760:	0796	90 DR	RCC	ZON	IF IT IS NOT SET
0770:	0798	A0 00	SETZ	LDYIM \$00	SWITCH TO 2400 HZ
0780:	079A	F0 D7	BEQ	ZON	ALWAYS
0790:					
0800:	079C	C6 F2	ROUT	DECZ	COUNT
0810:	079E	D0 CF	BNE	TRY	
0820:	07A0	60	RTS		ALL OVER GO HOME
0830:					

## TIMING TABLE

0840:					
0850:					
0860:	07A1	02	NPUL	=	\$02
0870:	07A2	C3	TIMG	=	\$03
0880:	07A3	03		=	\$03
0890:	07A4	7E		=	\$7E
0900:					
0910:					

0010:	07A5	A9 2F	TAIL	LDYIM	\$2F
0020:	07A7	20 6B 07	JSR	OUTCH	AS CHAR
0030:					
0040:	07AA	AD E7 17	LDA	CHRL	SEND
0050:	07AD	20 CB 07	JSR	OUTBT	CHECKSUM
0060:	07B0	AD E8 17	LDA	CHRH	LO AND
0070:	07B3	20 CB 07	JSR	OUTBT	HI
0080:	07B6	A2 02	LDYIM	\$02	AND SEND 2
0090:	07B8	A9 04	LDYIM	\$04	EOT CHARS
0100:	07BA	20 5F 07	JSR	HIC	
0110:					
0120:	07BD	AD 02 17	STPQ	LDA	\$1702
0130:	07C0	09 08	LDYIM	\$08	
0140:	07C2	8D 02 17	STA	\$1702	
0150:	07C5	4C 8C 1E	JMP	INIT	

0160:			*		
0170:			*	SUB TO SEND CHAR WITH CHECKSUM CALCULATION	
0180:	07C8	20 4C 19	OUTBT	JSR	CHRT
0190:					
0200:			*	SUB TO SEND BYTE AS TWO ASCII CHARS	
0210:	07CB	48	OUTBT	PRB	SAVE BYTE
0220:	07CC	4A	LSRA		GET
0230:	07CD	4A	LSRA		UPPER
0240:	07CE	4A	LSRA		NYBLE
0250:	07CF	4A	LSRA		
0260:	07D0	20 D4 07	JSR	HEXT	AND SEND IT
0270:	07D3	68	PLA		RETURN BYTE
0280:			*		
0290:			*	SUB TO SEND ONE HEX CHAR AS ASCII	
0300:	07D4	29 0F	HEXT	ANDIM	\$0F
0310:	07D6	C9 0A	CMPIM	\$0A	CHANGE TO ASCII
0320:	07D8	18	CLC		BY ADDING



# KIM SYSTEM SOFTWARE

0330: 0709 30 02  
0340: 070B 69 07  
0350: 070D 69 30  
0360: 070F 4C 64 07

HEXAT

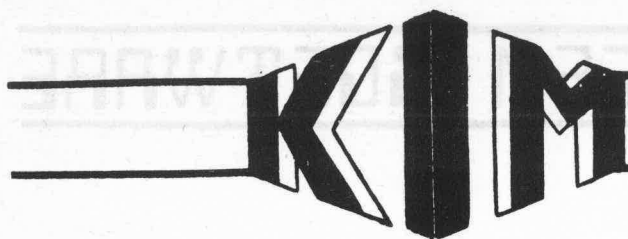
HMI HEXAT  
ADCIM \$07 37 TO A...F  
ADCIM \$30 AND 30 TO 0...9  
JMP OUTCHT

## SYMBOL TABLE 3200 3440

ACIA	1400	BIAD	06FF
BUFFER	0480	CHKH	17F8
COUNT	00F2	DES	05D3
EAH	17F8	EAL	17F7
FIRST	0566	GANG	00F0
GETLYN	06AF	GET	06D1
HICA	0761	HIC	075F
INALLR	057D	INCVER	19EA
INTVER	1932	LEADER	0733
MODEMI	0524	MODEMO	055F
NEXT	073D	NIN	0664
NOOP	0687	NORM	0518
ONEMIA	058D	ONEMIN	074C
OUTBT	0708	OUTIRC	0708
PAKM	05F2	PLAT	05B1
PRINTR	05A3	PRTHD	063F
RDRYTE	19F3	RDRHT	1A24
RED	05F8	RESHT	2031
SAL	17F6	SAL	17F5
SETZ	0748	SORCOT	07AC
SWITCH	0526	SYNC	0643
TIC	00F1	TIMG	07A2
TRIR	00FF	TRY	076F
WACHT	0592	WRITEQ	0534
ZONA	0777	ZON	0773

BIADNO	06F
CHKL	17E7
DOIR	06DF
ENDAD	06A4
GETCHR	06CD
HEXAT	070D
INALL	05DC
INIT	1F8C
LIDAR	0580
MODI	051B
NMOD	055E
NORMA	0574
OUTALA	057A
OUTCHT	076B
PLET	0642
PRTHDA	05AM
RDMP	05A0
RES	062F
SRD	1742
STPO	07BD
TAIL	07A5
TKSTX	06E4
IST	06BF
XTEMP	00F5

BUFFER	0400
CHKT	194C
DUMP	0704
EOFC	0759
GETLIN	068E
HEXT	07D4
INALLA	05DF
INOT	0604
MEMY	05DA
NEWLYX	06A1
NOCRA	074F
NPUL	07A1
OUTALL	0668
PAGIN	05D8
PLYT	05AA
RDRIT	1A41
READO	0507
ROUT	079C
SEE	055R
STRIX	0500
TAMTMP	05D9
TMP	00F3
VFR	17EC
YTMP	00F4



WAT DOE IK MET MIJN KIM

P.J. Visser

In deze aflevering als laatste redactie-lid de rij sluitend, zal ik een en ander vertellen over de kimmen waarme ik iets uit voer. Toen ik in 1975 voor het eerst de kim leerde kennen, was het eigenlijk al liefde op het eerste gezicht en begon ik al gauw, zoals de meesten waarschijnlijk, de meegeleverde dokumentatie te bestuderen om de KIM I aan de praat te krijgen.

Een van de eerste programmatjes die wel wilden werken, waren telprogrammatjes en spelletjes.

Toen deze periode voorbij was, kwam het moment, waarop serieus met plannen werd begonnen om zinvolle dingen te gaan doen met de kim. Toen ook is de idee ontstaan, de kim als uitgangspunt te kiezen voor een veel omvangrijker systeem. De geboorte van een T4 microcomputer-configuratie was een feit.

Al direkt in het begin werd de kim ingezet bij het in bedrijf stellen van nieuwe uitbreidingen op hardware gebied. Er kwam een bufferkaart, een 8K ram kaart, digitale in- en uitgangsmogelijkheden en later ook nog een ADC print.

Met grote inzet van de huidige Kimclub voorzitter C. Filmer en Siep de Vries, zijn in enkele maanden tijds diverse stukken en stukjes software en hardware tot stand gekomen, waaruit uiteindelijk het zgn. T4 systeem is ontstaan.

Mijn huidige KIM/T4 systeem bestaat uit: Kim 1, busbufferr, 32K ram, 8K rom, lezer/ponser interface, een burroughs lezer en ponser, een teletype, een ASCII Display terminal en een cassetteaansluiting.

Zoals reeds gezegd, gebruik ik mijn kimmen ( het zijn er nu reeds 3) grotendeels in mijn bedrijfje als zinvol gereedschap om bijvoorbeeld nieuwe printen te testen, storingzoeken in schakelingen, aanmaken van systeem software, dit laatste sinds kort toegevoegd aan het aktiviteitenpakket van het bedrijfje, mogelijk te maken.

Nu iets over de KIM in de komende maanden.

Zoals reeds bekend is, zal de KIM langzamerhand verdwijnen uit ons midden, daar de fabriek de produktie gaat beperken, waarschijnlijk zelfs geheel stop zetten.

Dit is een onzekere ontwikkeling, welke des te zwaarder gaat tellen, als je ervan uitgaat dat deze kaart een onmisbaar (?) onderdeel vormt in bestaande procesbesturingselektronika welke eerder werd gebouwd.

Daarom ben ik nu zover dat, misschien wat voorbarig, ik u de wedergeboorte van de KIM maar nu op eurokaartformaat kan aankondigen.

De nieuwe microcomputerkaart zal ET 40 gaan heten en de kaart ziet er als volgt uit: 6502 CPU (2 MHz versie), 2K statische ram, 2K EPROM type 2708, een PIA type 6821, en een ACIA type 6850 met kristalgestuurde bitrate-generator. De kloksignalen voor deze onderdelen komen uit een 4 MHz Xtal klok met delers naar 2 en 1 MHz.

Deze processorkaart zal het verjongde hart worden van een geheel van gedaante veranderde T4 systeemconfiguratie. Het nieuwe systeem gaat EURO-T4 heten en ik hoop u reeds op de volgende KIM club bijeenkomst een geheel werkend systeem te kunnen laten zien.

## PRINTER VOOR DE KIM-I

Dhr. Bicknese

Bij de firma Manudax te Heeswijk kan men een sympathiek metaalfolie-printertje kopen, welke uitstekend gebruikt kan worden voor de KIM. Het is de MP-300 printer, 32/64 tekens per regel, 64 tekens per sec. De printer wordt gestuurd met een 6 bits ASCII serie input signaal. In de monitor van de kim is reeds een volledig programma aanwezig, om de printer te sturen, zodat na het aansluiten van de printerinterface, welke niet behoeft te worden afgeregeld, het geheel direkt bedrijfsklaar is.

De interface uit figuur 2 is in staat uit de data signalen de besturing signalen te selecteren en te interpreteren.

Alle serieel binnenkomende bits worden parallel aan de printer aangeboden, tenzij het besturingssignalen zijn.

De stuursignalen zijn: (ASCII)

0D	(CR=carriage return)	Print current line
0A	(LF=line feed)	Skip one line
18	(CAN=cancel CTRL X)	Clear input buffer

Typt men -in de echo mode, zie verderop- op het keyboard de returntoets (enter) dan zal de current line worden geprint. Is de regel vol, dan zal de printer automatisch de current line printen.

De linefeed toets heeft tot gevolg, dat het papier één regel wordt opgeschoven. (Print spatierregel)

Geeft men CTRL X, dan wordt de inhoud van een regel gewist en kan men opnieuw intypen.

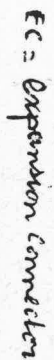
Nu doet zich echter de moeilijkheid voor, dat de monitor iedere regel afsluit met de codes 0D en 0A, wat tot gevolg heeft, dat na het printen van een regel het papier opschuift naar de volgende regel (0D code) en vervolgens nog een regel opschuift. (0A code)

Het resultaat is, dat iedere current line door een spatierregel wordt gevolgd. Om dit te ondervangen maakt de interface onderscheid tussen signalen, komende uit de monitor en komende van het keyboard.

De interface onderzoekt hiervoor de toestand van het key press signaal. De 0A code van de monitor wordt genegeerd daar deze niet vergezeld gaat van het keypress-signaal. Wil men nu via de software een regelopschuivi bewerkstelligen, dan kan hiertoe een extra 0D code worden gegeven.

Nu kan men met de ECHO toets op het keyboard beslissen of de data, ingetypt op het keyboard, direkt naar de printer wordt overgebracht of niet. De ECHO toets heeft geen invloed op signalen, komende vanuit de monitor. Met schakelaar S1 (dubbelpolig, de andere pool zit op de voedingslijn) kan de printer worden in- en uitgeschakeld.

Meer informatie:   MP-300 printer, L-type (metaalfolieprinter)  
                           Manudax Nederland BV  
                           Meerstraat 7  
                           5473 ZG Heeswijk (NB)   Tel: 04139- 1252





## KEYBOARD VOOR DE KIM-I

Dhr. Bicknese

Met behulp van de interface uit fig. 1 kan men op eenvoudige wijze een ASCII keyboard op de KIM aansluiten.

In de monitor is reeds een volledig programma aanwezig welke de binnenkomende signalen converteert en interpreteert.

De bijzonderheden hiervan vindt u in uw KIM handboek. De output van het keyboard is parallel; door de interface wordt deze serieel omgezet en op de keyboard input van de KIM aangeboden (aansluiting T op de applicatie-connector)

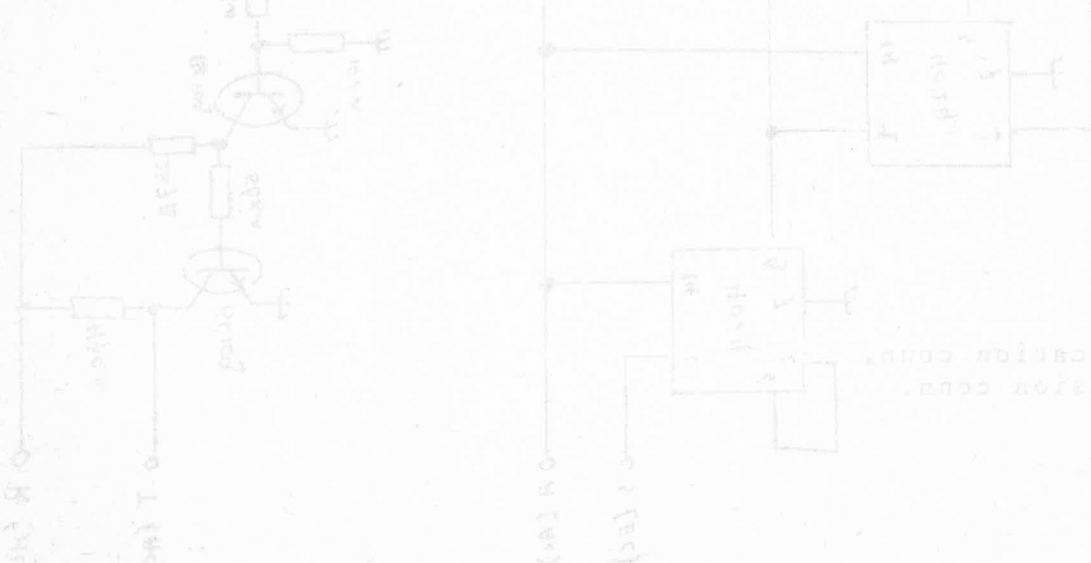
De interface is dusdanig van opzet, dat zij niet behoeft te worden afgeregeld.

Wordt op het keyboard een toets ingedrukt, dan zorgt het keypress signaal ervoor, dat op de output van de interface een startbit (logisch nul nivo) ontstaat. Intussen wordt op de parallel output van het keyboard de data klaargezet. Na het startbit wordt de data bit voor bit naar de KIM doorgeklokt. Ieder bit heeft dezelfde puls-breedte als het startbit.

Voordat de KIM "weet" dat er een keyboard is aangesloten, geeft met eerst een reset (wij hebben hier de blanco toets voor gebruikt) en daarna een backspace (rub out). De KIM weet nu de lengte van het eerste startbit om de overdrachtsnelheid van de bits te kunnen bepalen. Deze overdrachtsnelheid is hoog: de repeat snelheid van het keyboard (10 aanslagen per seconde na 1 sec. keypress) kan gemakkelijk worden gevolgd.

Wij gebruikten een KBD-5 keyboard met ASCII encoder type 219W.

Rhapsody, San Antonio Texas 78216, welke normaal in Nederland bij bekende verkoopadressen is te verkrijgen \*\*





AC = application conn.  
EC = expansion conn.

## SCHAKELING VOOR HET OMZETTEN VAN TTL NAAR RS 232 C en omgekeerd.

P.J. Visser

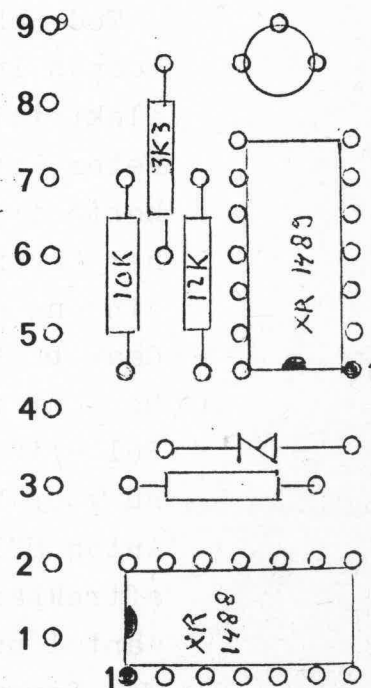
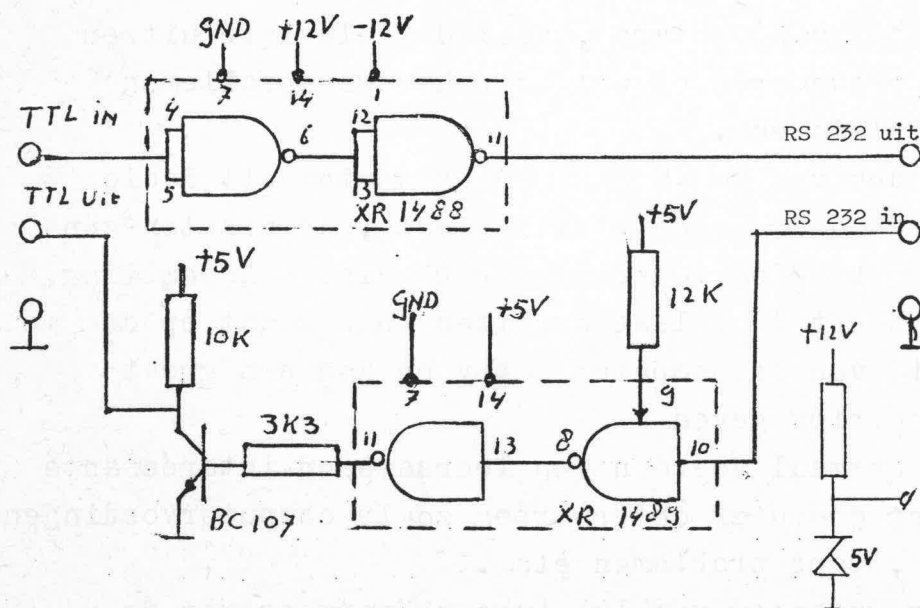
Teneinde het zo nu en dan weer ontstaan van "interface problemen" bij het aansluiten van microcomputers aan bijv. terminals, of andere apparaten welke een spanningsingang hebben volgens RS 232 C, op te lossen, is hieronder een eenvoudig schakelingetje afgebeeld.

Benodigheden: Een printje (is reeds ontwikkeld, zie elders in dit blad), een paar weerstanden, en de twee IC's 1489 en 1488 welke het uiteindelijke werk moeten doen.

Verder zorgt een zenerdiode van 5,1 volt voer de benodigde 5volt voeding, welke wordt afgeleid van de benodigde +12 en -12 volt. Een transistor type BC 107 zorgt ervoor, dat het op de 1489 aangeboden RS 232 signaal weer als TTL compatible aan de uitgang verschijnt.

Aansluitingen op de print:

1 = GND	4 = +5V	7 = TTL ingang
2 = -12V	5 = RS 232 ingang	8 = TTL uitgang
3 = +12V	6 = RS 232 uitgang	9 = GND



## Bijeenkomst 15 maart 1980 KIM club

Dateq Almere .J.C.Otten

Op het programma van deze dag stond een grote verscheidenheid aan lezingen en demonstraties .

Allereerst werd een modem verbinding gedemonstreerd tussen een Apple van Uwe Schröder en de KIM van Willem van Gelderen , deze KIM stond bij Willem thuis . De modem werkte goed en de demonstratie interessant .

Een daarmee samenhangend onderwerp werd door Rinus Vleesch-Dubois behandeld, namelijk het aansluiten van een ACIA 6850 aan de computer om RS232 interface mogelijk te maken .

Over de modem en de 6850 zal nog wel meer gepraat en geschreven worden . Een introductie in de programmering van de ACIA 6850 werd door Uwe Schröder gegeven .

Ted Schouten vertelde het een en ander over de toepassing van de microcomputer bij het bewaken van de elektriciteitsvoorziening bij de PEN .

Peter Visser vertelde hoe hij met zijn bedrijf in zeer korte tijd de hard en software voor het telebingospel , het bekende spel van de Avro , voor de televisie uitzenden dingen moest produceren en wat voor hardwareproblemen daar bij kwamen kijken .

De achtergronden van de 4K RAM geheugenprint uit Radio Bulletin werden door Hans Otten belicht , de printen zijn nu voor KIM club leden verkrijgbaar bij Visser Assembling.

Anton Müller heeft de belastingwetten onderzocht op de aftrekbaarheid van de computer hobby en kon een groot aantal nuttige tips geven .

Een forum en de zaal voerden een leerzame en interessante discussie over computer onderwerpen zoals computervoedingen en storingen , tape problemen etc .

Demonstraties waren er van Ing.bureau Koopmans met de kleurenversie van de Challenger 4P en Uwe Schröder met Apple's



## D A T A - COMMUNICATIE.

P.J. Visser

Een samenvatting van de door Siep de Vries gehouden lezing over datacommunicatie op de KIM-club bijeenkomst te Krommenie, 19 jan. 1980.

### Inleiding.

Wanneer we gegevens van een bepaalde plaats naar een andere plaats willen overbrengen, komen we al gauw tot de ontdekking dat dit niet zo eenvoudig is. Neem bijvoorbeeld een verbinding tussen twee apparaten. Hier reeds zien we, dat er kabels met stekers en soms schakelaars aan vast zitten, waar onderbrekingen kunnen ontstaan. Gaan we nu naar een PTT lijn (telefoon) kijken, dan komen we onder andere pluggen, soldeerverbindingen (!), schakelkontakten in centrales, versterkers en verzwakkers tegen. Als we bijv. van Alkmaar naar Heerhugowaard "DATA" willen sturen, zijn er vele onderbrekingen mogelijk. Het gevolg van deze "obstakels" is, dat de over te brengen signalen zullen worden vervormd. Deze vervormde signalen zijn het probleem van de datacommunicatie. Het resultaat hiervan is, dat overgeseinde data mogelijk verminkt zal aankomen en voor een tweede, zo niet een derde keer opnieuw zal moeten worden verzonden. DIT KOST EXTRA TIJD en dus EXTRA TELEFOONLIJN-KOSTEN. Een auto met chauffeur zou in een dergelijk geval een meer betrouwbare vorm van vervoer kunnen betekenen waar het data betreft.

Hoe ondervangen we deze verminkingen van data nu?

Het antwoord is eenvoudig. Ze zijn niet te voorkomen.

Wat kunnen we er aan doen, zal men vragen. Het antwoord is al even eenvoudig, namelijk "VOLLEDIGE RECOVERY" ofwel zorgen dat er zoveel "herkansingen" plaats vinden als nodig is om de data uiteindelijk geheel foutloos op de plaats van bestemming te doen aankomen.

Hiertoe heeft men in de computerwereld het aloude begrip PROTOCOL wederom ter hand genomen, wat zoveel wil zeggen als: zo hoort het.

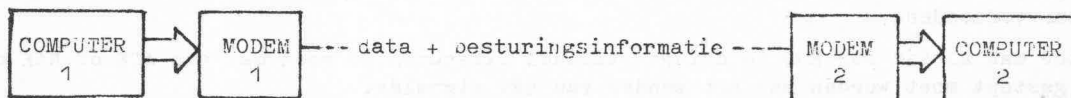
Er zijn eenvoudige protocollen, zoals bijv. de kim hanteert met zijn parity check en/of checksum. Er zijn ook ingewikkelder protocollen als IBM 2780, BISYNC, SDLC/HDLC en andere.

### Verbindingsapparatuur.

Wanneer we over datacommunicatie-netwerken praten, zullen we spoedig te maken krijgen met randapparatuur, welke deze communicatie moet verzorgen, de zgn. MODEM ofwel modulator-demodulator.

Een modem is in het algemeen een apparaat dat data verpakt, en wel zodanig dat deze bij verzenden zonder kleerscheuren op de plaats van bestemming aan komt.

Nu eerst iets over deze verpakking.



### Teletype protocol.

Het meest simpele protocol is het verzenden van de ASCII-code (8-bits) aangevuld met twee characters, bedoeld om bijv. de lezer of ponsler van de TTY aan of uit te zetten.

Bekend zijn bijv. ↑Q, ↑S, ofwel control Q, control S.

↑Q = lezer aan

↑S = lezer uit

We krijgen echter geen bevestiging terug van de teletype, of de weggezonden data werkelijk geponsd is.

Om deze bevestiging wel te kunnen verkrijgen, stappen we over naar de zgn. duplex methode.

DUPLEX METHODE.

We kennen o.a. de begrippen HALF DUPLEX en FULL DUPLEX.

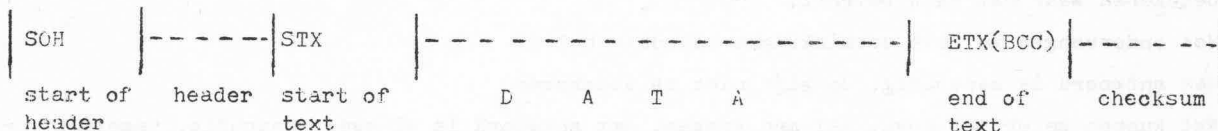
Om een full duplex verbinding tot stand te brengen, moet men de beschikking hebben over twee telefoonlijnen. Dit is een vrij kostbare methode, denk aan PTT lijnkosten. Wij zullen ons dan ook bezig houden met de zgn half duplex verbinding, waarvoor slechts één telefoonlijn nodig is.

Half duplex verbinding: Toegepaste transmissiesnelheden: 1200 Baud heen,  
75 Baud terug.

Standaardisatiegegevens hierover kunnen we vinden in de ECMA 16 en ECMA 41 standaard.

ECMA 16 De standaard ecma 16 is een zgn byte-protocol. Er zijn 255 bitcombinaties als dataword mogelijk.  
ECMA 16 bemoeit zich alleen met de ASCII characters welke geprint kunnen worden, dus niet alle ASCII tekens vallen hieronder. Communicatie characters, welke bijv. niet in de data mogen zitten, zijn o.a. STX (start of text)  
ETX (end of text)

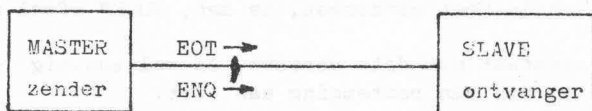
Een over te zenden bericht zal er nu als volgt uit zien:



Verder is als controle bekend de EVEN PARITEIT voor asynchrone - en de ONEVEN PARITEIT voor synchrone datatransmissie.

Na het uitsturen van een bericht kan bij half duplex systemen een antwoord worden verwacht zoals bijv.:

ACK	=	acknowledged	=	begrepen
NAK	=	not acknowledged	=	niet begrepen
EOT	=	end of text	=	einde bericht
NIETS	=	???	=	wat nu? (time-out maken van bijv. x sec)



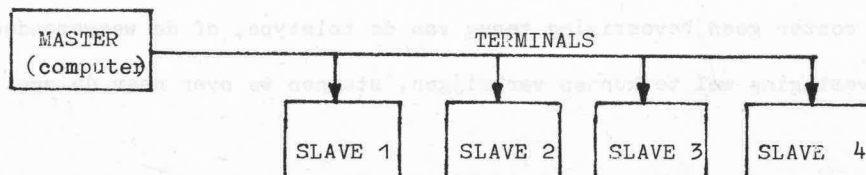
Een EOT signaal wordt gestuurd om de master-slave functie om te keren en wordt altijd door de master verzonden.

De master mag altijd een ENQ (enquiry) signaal uitsturen en moet dan een ACK of NAK ontvangen, waarna gestopt moet worden met het zenden van ENQ signalen.

Hoe starten we nu met twee computers welke allebei wel iets naar elkaar toe willen sturen. Allereerst dienen we te weten, wie de master en wie de slave is. Hiertoe is een neutrale stand beschikbaar, waarin het systeem komt te staan na enige malen doelloos te hebben staan schakelen tussen master en slave, synchroon met de andere kant van de lijn.

Dit betekent, dat dan automatisch een keuze wordt bepaald, en beide systemen verschillend staan geschakeld, nodig om data te versturen.

Moeilijker wordt het, indien er een master (computer) is en meerdere slaves, bijv. data terminals



In dat geval krijgt ieder station zijn eigen adres.

Zodra nu een ENQ signaal wordt gestuurd, wordt een adres toegevoegd om het gewenste station te selecteren.

Het geadresseerde station dient dan een ACK of NAK terug te zenden. Dit wordt "POLLING" genoemd. Deze vorm van communicatie wordt ook wel "MULTI-DROP-LINE" genoemd.

Een uitbreiding om meer mogelijkheden te benutten.

Als de data als ASCII tekens wordt verzonden, zijn er maar een beperkt aantal tekens (max. 128). Naar terminals met een meer uitgebreide karakterset (grafische display terminals) kan door in de datastroom de code DLE ETX op te nemen erop worden geattendeerd dat de volgende ASCII karakters een andere betekenis hebben.

Beperken van transmissie-fouten en data verminking.

Stel, u wilt een bericht verzenden, dat 60.000 karakters groot is, met een snelheid van 30 characters per seconde.

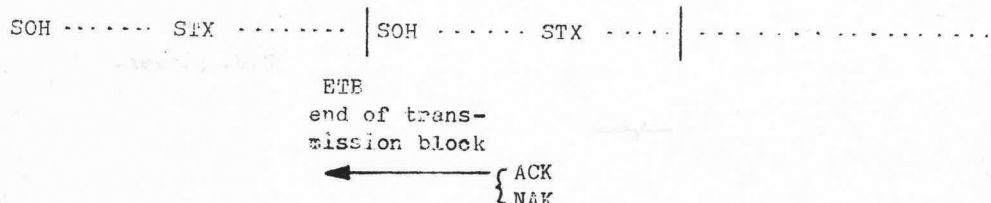
Het overzenden van dit bericht zal 2000 seconden in beslag nemen.

Nu krijgt men als antwoord van het ontvangende station "NAK" (niet begrepen)

We proberen het voor de tweede keer nog eens. Het terugkomende antwoord: NAK en weer moet het tijdrovende overzenden van het gehele bericht beginnen.

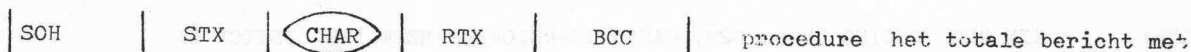
Snellere methoden.

We delen een groot bericht op in kleinere stukken en zenden deze apart uit. Zo besparen we kosten en tijd (PTT lijnen zijn ook kostbaar). En tevens weten we sneller of een van de stukken data niet goed is aangekomen, dus bijv.:



Nadeel van dit protocol en van de meeste andere overigens, is de "overhead"

Als n.l. een bericht een karakter lang is, dan wordt volgens de

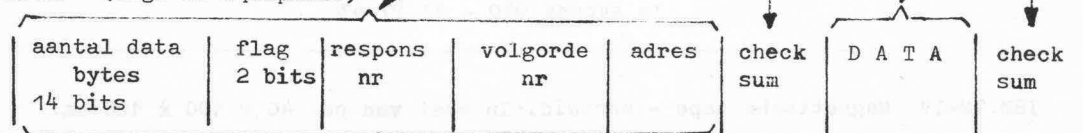


enveloppe 5 characters lang, om één karakter te verzenden!

Om deze "overhead" te beperken, hebben we op terminals vaak de zgn. "block-mode". Nadeel hiervan is weer, dat 1 karakter wijzigen op het scherm als konsekwentie heeft, het opnieuw uitsturen van de gehele beeldscherm inhoud.

Interessante protocollen.

DDCMP (Digital Equipment)



Dit protocol is bedoeld voor het gebruik van meerdere "lijnen" tussen twee computers.

Het regelt het verkeer van berichten zodanig dat aan de ontvangende zijde de berichten uiteindelijk in dezelfde volgorde worden verwerkt als waarin ze zijn uitgezonden.

Dit, ondanks de mogelijkheid, dat een kort bericht een lang bericht kan "inhalen", indien beide berichten niet over dezelfde lijn worden verstuurd.

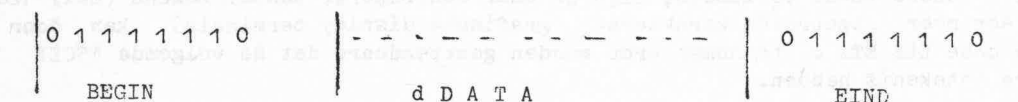
Hiertoe is dan ook als extra hulp het zgn. volgorde nr. een onderdeel van dit protocol.

Het zgn. respons nr., ook onderdeel van dit protocol, wordt gebruikt om informatie over reeds ontvangen berichten terug te zenden naar de afzenden. Dit bespaart dan weer op de bezettingstijd van de lijn, waarover de berichten worden verzonden.

# DE KIM KENNER

## SDLC (IBM)

Een SDLC protocol ziet er als volgt uit:



Indien de zender 5 maal een 1 ziet, maakt hij er een extra nul tussen, zodat het begin en eind-sigitaal "unieke" signalen blijven in het overgezonden bitpatroon!

Als de ontvanger nu vijf enen en een nul ziet, gooit hij dit 6<sup>e</sup> (de nul) weg, zodat het oorspronkelijke data sigitaal weer hersteld wordt.

Indien 8 x een 1 wordt gestuurd, betekent dit "STOP" .

Als u meer informatie zoekt over "datacommunicatie" dan kunt u de volgende boeken wellicht eens raadplegen of aanschaffen:

" Technical aspects of Data Communication" door John. E. mC. Namara, verkrijgbaar bij:

Digital Equipment Corporation te Utrecht. Prijs is ca. f 75,00

P.J. Visser.



---

AANGEBODEN: KIM MET VOEDING 5V en 12V, CASSETTE-RECORDER MERK "REALISTIC" EN

MICRO-CHESS SOFTWARE CASSETTE. PRIJS N.O.T.K.

Te bevragen bij: J.G. Klein,

Koning Arthurlaan 16

Eindhoven

Telefoon: overdag 040 - 11 62 62

's avonds 040 - 43 27 43

---

AANGEBODEN: IBM 72-IV Magnetische tape - eenheid. In kast van ca. 40 x 100 x 120 cm.

Prijs n.o.t.k. Eventueel ruilen voor microsoft 8K basic ook mogelijk.

Belangstelling? Bel mij dan even.

P.J. Visser,

Toussaintstraat 7

1814 EG Alkmaar

Tel: 072 - 12 66 52



# KIM VRAAG EN AANBOD

## RUBRIEK VRAAG EN AANBOD

### Aangeboden :

IBM vertolkende kaartponsmachine 029  
vraagprijs f 4000,-

A. Müller tel.020-860245

### Gevraagd :

KIM-1 gaarne opgave van prijs

K. Hermanides Achterbosk 15 9036 KV Menaldum  
telefoon na 19.00 uur 05185 - 702

### Gevraagd :

IBM schijvenunit 2311

A. Müller tel. 020-860245

### Aangeboden :

- Blattspeicher met 2 cassette's capaciteit 1  
toegangstijd 850 ms . In fraaie kast met voeding en  
electronica prijs f 750,-
- IBM schrijfmachine voor bovenstaande set bruikbaar ,  
geschikt voor computer I/O ( 24V ) prijs f 650,-
- 8gats ponsband maker merk Olympia f 75,-

### Aangeboden :

SYM-1 met 4K RAM , telexinterface met monitor ,  
ASCII toetsenbord in kast , videointerface met rf modu-  
lator , voeding 5 V 4A ,12 V 0,5 , -12 V 0,5A , 24 V 0,1A  
Alles in een koop f 1200,-

E. Bledoeg tel. 070 - 604071

### Aangeboden :

9 stuks UV EPROM 1702Q nieuw ongebruikt t.e.a.b.

H.J.C. Otten tel.02940 - 13349 vragen naar Hans

17 MEI 1980 BIJEENKOMST KIM GEBRUIKERS CLUB  
NEDERLAND. PLAATS: WORDT NADER BEKEND GEMAAKT.

19 - 22 MEI 1980 NCC WORDT DIT JAAR GEHOUDEN IN ANAHEIM.

19 - 21 JUNI 1980, PRAAG CZSSR, TC5 IFAC/IFIP REAL  
TIME PROGRAMMING WORKSHOP. INL.: IFIP SECR.,  
3 RUE DU MARCHE, CH-1204 GENEVE, ZWITERSLAND.

23 - 27 JUNI 1980, CAMBRIDGE G.B.: SYMPOSIUM ON RESEARCH  
AND DEVELOPMENT IN INFORMATION RETRIEVAL.  
INL.: C. J. VAN RIJSBERGEN, COMP. LAB.,  
CORN EXCHANGE STREET, CAMBRIDGE CB2 3QB, G.B.

23 - 27 JUNI 1980, ROME, ITALIE: IBI WORLD CONFERENCE  
ON TRANSBORDER DATA FLOW POLICIES. INL.: IBI, VIALE  
CIVILLA DEL LAVORO 23, POB 10253, 00144 ROME, ITALIE.

23 - 27 JUNI 1980 KONFERENTIE APL80 IN HET LEEUWEN-  
HORST CONGRES CENTRUM TE NOORDWIJKERHOUT. INL.: CRI,  
POSTBUS 9512, LEIDEN.

25 - 27 JUNI 1980, INTERLAKEN, ZWITSERLAND:  
SIMULATION 80: 3RD INTERNATIONAL SYMPOSIUM.  
INL.: SIMULATION, POB 354, CH-8003 ZURICH, ZWITSERLAND.

26 - 28 JUNI 1980, SZEGED, HONGARIJE: 2ND INDUSTRIAL  
ROBOT COLLOQUIUM. INL.: SCIENTIFIC SOCIETY OF MECHANICAL  
ENGINEERS, POB 451, H-1372 BUDAPEST, HONGARIJE.

30 JUNI - 11 JULI 1980: HERTFORD, G.B.: OPTIMISATION,  
TECHNIQUES AND APPLICATIONS. INL.: PROF. L C W DIXON  
THE NUMERICAL OPTIMISATION CENTRE, HATFIELD POLYTECHNIC  
HATFIELD (HERTS) AL10 9AB, G.B.

2 - 5 JULI 1980, AMSTERDAM. ASIB 80: 4TH CONFERENCE  
ON ARTIFICIAL INTELLIGENCE. INL.: DR. B. WIELENGA,  
PSYCH. LAB., WEESPERPLEIN 8, AMSTERDAM.

29 AUGUSTUS - 7 SEPTEMBER 1980 FIRATO, RAI, AMSTERDAM.

20 SEPTEMBER 1980 BIJEENKOMST KIM GEBRUIKERS CLUB  
NEDERLAND. PLAATS: WORDT NADER BEKEND GEMAAKT.

3 - 22 OKTOBER 1980 EFFICIENCY BEURS, RAI, AMSTERDAM.

3 - 7 NOVEMBER 1980 FIAREX, RAI, AMSTERDAM.

15 NOVEMBER 1980 BIJEENKOMST KIM GEBRUIKERS CLUB  
NEDERLAND. PLAATS: WORDT NADER BEKEND GEMAAKT.

19 - 24 NOVEMBER 1980 MICRO EXPO TE PARIJS.



APPLE

UCSD - PASCAL

f 1270 ex / 1498,60 incl

compleet met

Pascal kaart met 16k RAM en AUTOSTART ROM

Pascal handboeken en leerboeken

Pascal diskettes (filer, editor, compiler, linker, macroassembler, library)

Basic handboeken / leerboek en Integer / Applesoftbasic op diskette

APPLE II, 48 uur getest ..... 3060 ex / 3611 incl btw

, per geleverde 16k RAM - uitbreiding ..... 295 ex / 348 incl btw

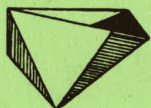
Mini - floppy incl controller, DOS, handboek ..... 1750 ex / 2065 incl btw

MACRO-assembler, text editor voor PET/APPLE/??? ..... 138 incl btw

prijswijzigingen voorbehouden

**INGENIEURSBUREAU SCHRÖDER**

Echternachln 161, 5625 KC Eindhoven, 040-421821



**VISSER ASSEMBLING ELECTRONICS BV**

Postbus 426

-

1800 AK Alkmaar

-

Tel. 072-126652

PROGRAMMERING VAN 6502 EN 6800 SYSTEMEN

VERZORGEN VAN BESTURINGS- EN REGELELEKTRONIKA

PRINTKAARTEN ONTWERPEN EN VERVAARDIGEN

ONDERDELENLEVERANTIE VOOR AMATEUR EN PROFESSIONAL

ADVIEZEN EN REALISATIE IDEEEN VOOR INDUSTRIE-ELEKTRONIKA

BESTÜCKEN EN MACHINESOLDEREN VAN PRINTSERIES

TESTEN VAN PRINTKAARTEN M.B.V. MICROCOMPUTERS

KOMPONENTEN ONDERZOEK EN PROTOTYPE-ONTWIKKELING

SPECIAAL-ELEKTRONIKA, OP DE KLANT AFGESTEMD, OOK KLEINE SERIES.

P8004-4